

FINANCIAL DATA FORECASTER

Supervisor: Dr. Beta Yip

Karan Mahajan (3035346025)

Final Report

03 MAY 2020

ABSTRACT

Financial instruments’ price prediction has been an area of interest for many researchers around the world due to the amount of volatility and uncertainty associated with it. A considerable number of models have been developed in the past few years to understand the movement of stocks and perform predictions. Although several complex prediction strategies like Artificial Neural Networks, Radial Basis Function Neural Network model, Dirichlet Processes Mixture model, etc have been discussed in literature, studies seldom incorporate time series data from more than one market and consequently, lose out on valuable information. However, a handful of studies have proposed novel strategies that rely on the correlations between various markets or domain of information. Therefore, the project explores ways to effectively employ the correlations between various domains of financial markets like forex rates, stock prices, market indices, etc and market sentiment inferred by analysing posts on social media platforms for accurate predictions. It aims to do so by involving techniques like walk-forward forecasting, hyperparameter optimization and feature selection. Market data and social media tweets are scrapped from the world wide web and will act as raw input. The data was studied and analysed for stationarity and correlations. Feature engineering and experimental data analysis are the key steps of this process. Correlations between cross market data was exploited in this research and showed improved results. Support Vector Machine and Ridge Classifier showed superior performance over other traditional time series models. This research showed that the using hyperparameter optimization in conjunction with feature selection and walk-forward forecasting significantly improves on trading strategy in terms of overall rate of return and maximum downturn witnessed.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr Beta Yip, without whose support and guidance, this report would not have been possible. I would also like to thank Mr George Mitcheson for his feedback and suggestions on the project’s progress. I would also like to thank my CAES 9542 professor, Mable Choi for her insightful, inspiring and enjoyable teaching. The CAES course provided meaningful insights for effective report writing have been used throughout this report.

I would also like to thank my parents and my family for supporting me and helping me reach my full potential by allowing me to pursue my studies at The University of Hong Kong.

TABLE OF CONTENTS

[**ABSTRACT** i](#_Toc39339251)

[**ACKNOWLEDGEMENTS** ii](#_Toc39339252)

[**LIST OF FIGURES** v](#_Toc39339253)

[**LIST OF tables** vii](#_Toc39339254)

[**Chapter 1 Introduction** 1](#_Toc39339255)

[1.1 Background and Motivation 1](#_Toc39339256)

[1.2 Objective 1](#_Toc39339257)

[1.3 Scope 2](#_Toc39339258)

[1.4 Significance of the Study 2](#_Toc39339259)

[1.5 Outline 3](#_Toc39339260)

[**Chapter 2 LITERATURE REVIEW** 4](#_Toc39339261)

[2.1 Linear Models 4](#_Toc39339262)

[2.2 Sentiment Analysis 4](#_Toc39339263)

[2.3 Machine Learning 5](#_Toc39339264)

[2.3.1 Deep Learning 5](#_Toc39339265)

[2.3.2 Cross Domain Analysis 5](#_Toc39339266)

[2.4 Time Series Analyses 7](#_Toc39339267)

[2.5 Summary 8](#_Toc39339268)

[**Chapter 3 METHODOLOGY** 9](#_Toc39339269)

[3.1 Introduction 9](#_Toc39339270)

[3.2 Programming Language 9](#_Toc39339271)

[3.3 Software Development Practices 9](#_Toc39339272)

[3.4 Data Preparation: Step-by-Step Approach 10](#_Toc39339273)

[3.4.1 Data Collection 10](#_Toc39339274)

[3.4.2 Data Processing 13](#_Toc39339275)

[3.4.4 Exploratory Data Analysis 13](#_Toc39339276)

[3.4.5 Feature Engineering 15](#_Toc39339277)

[3.5 Iterative Algorithm Development 15](#_Toc39339278)

[3.5.1 Test Train Split 16](#_Toc39339279)

[3.5.2 Parameter Selection 16](#_Toc39339280)

[3.5.3 Training and testing 17](#_Toc39339281)

[3.5.4 Evaluation Metric and Techniques 18](#_Toc39339282)

[3.5.5 Back testing 19](#_Toc39339283)

[3.6 Optimization 21](#_Toc39339284)

[3.6.1 Walk Forward Classification 21](#_Toc39339285)

[3.6.2 Multithreading 22](#_Toc39339286)

[3.6.3 Feature Selection 23](#_Toc39339287)

[3.7 Summary 24](#_Toc39339288)

[**Chapter 4 PROGRESS, EXPERIMENTS AND RESULTS** 25](#_Toc39339289)

[4.1.1 Numerical data 25](#_Toc39339290)

[4.1.2 Textual data 26](#_Toc39339291)

[4.2 Data Processing 27](#_Toc39339292)

[4.2.1 Sentiment Analysis 28](#_Toc39339293)

[4.2.2 Processing Numerical Data 28](#_Toc39339294)

[4.3 Machine Learning Experiments 31](#_Toc39339295)

[4.3.1 Regression Models 31](#_Toc39339296)

[4.3.2 Time Series Models 33](#_Toc39339297)

[4.3. Classification Analyses 35](#_Toc39339298)

[4.4 Optimization 38](#_Toc39339299)

[4.4.1 Hyperparameter selection 38](#_Toc39339300)

[4.4.2 Sliding Window Approach 39](#_Toc39339301)

[4.4.3 Ensemble Voting 41](#_Toc39339302)

[4.5 Backtesting Strategies 41](#_Toc39339303)

[4.5.1 Simple Moving Average Strategy 42](#_Toc39339304)

[4.5.2 Intraday Trading Strategy 43](#_Toc39339305)

[4.6 Summary 45](#_Toc39339306)

[**Chapter 5 CONCLUSIONS** 47](#_Toc39339307)

[5.1 Major Conclusions and Results 47](#_Toc39339308)

[5.1.1 Conclusions from Machine Learning Models 47](#_Toc39339309)

[5.2 Further Research 50](#_Toc39339310)

[5.4 Work Distribution 51](#_Toc39339311)

[**Chapter 6 REFERENCES** 52](#_Toc39339312)

LIST OF FIGURES

[Figure 2.1 Correlation Between Multiple Financial Markets and Domains [2] 6](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339204)

[Figure 2.2 System Diagram of Ao's Proposed Cybernetic Mechanism [3] 7](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339205)

[Figure 3.1 Process of Data Preparation Involving The Following Steps: Data Collections, Data Processing, Sentiment Analysis, Exploratory Data Analysis, Sentiment Analyses And Feature Engineering 10](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339206)

[Figure 3.2 Web Scrapping Process Using a Web Scraping Application 12](#_Toc39339207)

[Figure 3.3 Seasonal Decomposition of Close Values 14](#_Toc39339208)

[Figure 3.4 Algorithm Development Lifecycle 16](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339209)

[Figure 3.5 Evaluation Metrics for Regression 18](#_Toc39339210)

[Figure 3.6 Evaluation Metrics for Classification 19](#_Toc39339211)

[Figure 3.7 Slow Moving Average Backtesting Strategy 20](#_Toc39339212)

[Figure 3.8 Algorithm for Walk Forward Classification 22](#_Toc39339213)

[Figure 3.9 PCA Analysis of BDT. Image Shows That Majority of The Variance is Explained by 8 Components for BDT. 23](#_Toc39339214)

[Figure 4.1 Most Recent Data Available for NASDAQ 100 26](#_Toc39339215)

[Figure 4.2 A JSON Object Containing Information About the Tweet Retrieved from Twitter 27](#_Toc39339216)

[Figure 4.3 Statistically Derived Features from Sentiment Analysis 28](#_Toc39339217)

[Figure 4.4 Seasonal Decomposition of Return on Hang Seng Index Close 29](#_Toc39339218)

[Figure 4.5 ADF Test Results Highlighting Stationarity of Derived Features. 30](#_Toc39339219)

[Figure 4.6 Correlation Matrix for Currencies for Time Period 2010-2020 30](#_Toc39339220)

[Figure 4.7 Predicted Raw Values by Linear Regression Model 32](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339221)

[Figure 4.8 Linear Regression Results on Raw Values of BDT 32](#_Toc39339222)

[Figure 4.9 Regression Graph for BDT Close Return 32](#_Toc39339223)

[Figure 4.10 ARIMA Diagnostics 33](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339224)

[Figure 4.11 ARIMA Regression Results Using Raw Values of BDT 33](#_Toc39339225)

[Figure 4.12 ARIMA Returns Forecast for BDT 34](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339226)

[Figure 4.13 ARIMA Fit Diagnostics Using Returns 34](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339227)

[Figure 4.14 Prophet Regression Results Using Raw Values for BDT 34](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339228)

[Figure 4.15 Prophet Regression Results Using BDT Returns 34](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339229)

[Figure 4.16 Classification Report for random Forest 37](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339230)

[Figure 4.17 Confusion Matrix for Random Forrest Classification on BDT 37](#_Toc39339231)

[Figure 4.18 Confusion Matrix For Ridge Classifier for BDT 37](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339232)

[Figure 4.19 Classification Report for Ridge Classifier for BDT 37](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339233)

[Figure 4.20 BDT Ridge Classifier Optimization 39](#_Toc39339234)

[Figure 4.21 Accuracy Results Across Different window Sizes 40](#_Toc39339235)

[Figure 4.22 Ensemble Voting Results 41](#_Toc39339236)

[Figure 4.23 Final Equity Graph for MNT 43](#_Toc39339237)

[Figure 4.24 Final Equity Graph for BDT 43](#_Toc39339238)

[Figure 4.25 Changes in Equity for BDT With Less Constraints 43](#_Toc39339239)

[Figure 4.26 Changes in Equity for MNT With Less Constraints 43](#_Toc39339240)

[Figure 4.27 Changes in for BDT 45](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339241)

[Figure 4.28 Changes in Equity for MNT 45](https://d.docs.live.net/1b841d3d0b6ede16/Documents/Final%20Report.docx#_Toc39339242)

[Figure 5.1 Sentiment Score for Dhaka Compared with BDT Close Return 49](#_Toc39339243)

LIST OF tables

[Table 1 List of Forex and Indices Chosen Along with Correlated Markets 31](#_Toc39017765)

[Table 2 MSE Scores for ARIMA And Prophet 35](#_Toc39017766)

[Table 3 Best Set of Hyperparameters Identified on an 80-20 split for BDT 38](#_Toc39017767)

[Table 4 Optimized Window Period for Initial Strategy 42](#_Toc39017768)

[Table 5 Optimized Window Sizes for the New Strategy 43](#_Toc39017769)

[Table 6 Division of Work 51](#_Toc39017770)

# **Introduction**

## 1.1 Background and Motivation

In recent years, financial forecasting has been a topic of interest for many researchers in the field of computer science. Moreover, the reduction in cost of computational power has acted as a catalyst and has created new opportunities for the use of technology in finance. The recent trends of globalization and mass interlinking of businesses globally has further resulted in an increase of trade around the world. Therefore, accurately predicting financial markets can be highly profitable in business organizations (ex. investment banks) where forecasting remains one of the most important activities in forming the basis for strategic and operational decision making [1].

The financial analysis of markets on a global scale is extremely complex and includes several subtle details [2]. This can be attributed to an increase in cross border trade which leads to markets influencing each other on a global scale. The ongoing trade war between the U.S. and China, effects of which are rippling through the global economy is but one such case of market interactions and its magnitude. Therefore, it can be concluded that a financial downturn in one country can have a domino effect on other countries. However, despite the abundantly evident correlation between markets, there is a scarcity of research that deals with the cross-market and cross domain analysis of financial data. Researchers having traditionally only dealt with historical data of financial instruments as a parameter to predict its future price [2] [3]. Moreover, diplomatic relationships between nations and political sentiment further affect global trade [4]. Thus, factors like sentimental analysis also need to be considered while creating models to forecast future market movement. In view of this, the aim of this project is to find relationships between multiple markets, with a primary focus on the Asia Pacific region, by the use of market indices, sentiment analysis, foreign exchange rates etc. as factors in order to accurately predict the dynamic movement of the indices of respective markets across the region.

## 1.2 Objective

The study aims to glean insights into whether traditional machine learning models could successfully forecast financial time series data. It also aims to discover whether there is a significant relation between markets contingent upon sentiment and market indices that may be of interest. Accordingly, it plans to investigate so by performing the following:

* Analyse time series of historical financial data (Index, foreign exchange, etc) and determine the combinations of most effective indicators as input.
* Identify trends and relationships between financial data of different markets.
* Analyse model behaviour to infer importance of features in predicting time series data.
* Identify relationships between the time series’ of financial data, market indicators and non-numerical data and achieve a favourable accuracy rate for financial predictions.
  + An accuracy rate of 70% for up-down movement of financial instrument and 60% for price bucket may be considered favourable.

## 1.3 Scope

To keep the scope of this project manageable, the project deals with markets in the Asia Pacific (APAC), apart from the United States of America (USA) which standardizes the exchange rates of markets in APAC. With each iteration of model development, trading and testing, the scope is narrowed down until four markets are left. The remainder of the project focuses on optimizing models to fit the data better, leading to improved accuracy and prediction quality. For our algorithm, we begin by expanding upon linear models of machine learning. Subsequently, models of greater complexity are employed to investigate the cross market as well as the cross-domain relationships of markets in order to increase our prediction accuracy.

## 1.4 Significance of the Study

It is anticipated that through analysing trends in market data and correlating it market sentiment derived from significant news and tweets we will be able to investigate the relationships between different markets in Asia Pacific region such as Bangladesh, Vietnam, Sri Lanka, Malaysia, etc This project will, therefore, illustrate how machine learning can be employed to achieve the aforementioned. Furthermore, the insights of this study can be used to make better strategic and operational decisions for businesses around the region.

## 1.5 Outline

The report is divided into 5 chapters. Chapter 1 has provided the background and motivation behind the study. It has also elaborated on the objectives, scope and significance of the study. Chapter 2 discusses the various related works. It summarizes the various historical research related to this field in a chronological manner. It begins by illustrating the traditional approaches. It then introduces the literature that discuss models developed by incorporating sentimental analysis and then culminates with the most recent advancement in this field, i.e. cross domain analysis of financial data. Chapter 3 of this report gives an overview of the methodology we plan to use in this project. It includes a description of how we plan to collect data, perform analysis and engineer features. Further, a description of how we plan to perform analysis on this data and develop models for predictions is also provided. Chapter 4 presents the progress made so far. Description of project website, data collected, and challenges faced are presented here. Chapter 5 concludes this report. Major conclusions made so far are mentioned here. The report will end with a reflection on the future work planned for this project.

# **LITERATURE REVIEW**

Historically, researchers and developers have employed historical time series of financial data as the sole parameter in data forecasting. Simple models like linear regressions, Hidden Markov Models, moving average, etc were employed to predict future trends. Recently, researchers have started focusing on determining hidden relationships between various other parameters, like market sentiment, political sentiment, market indices etc, that could inevitably affect the future market trends. This section discusses the forecasting techniques that have previously been used to predict financial data.

## 2.1 Linear Models

Autoregressive moving average is one of the models that have typically been used to analyse stationary time series data. This method takes past values, prediction errors and a random term into account while making predictions [5]. Other methods like logistic regression [6] and Hidden Markov Models [7], which check for any systematic repeating patterns in time-series data, have also been used to make predictions. However, these models have been unsuccessful to capture the complex non-linear relationships between financial markets and have yielded unfavourable results [2]. Thus, linear models have predominantly been the starting point in recent researches which eventually move on to more complex algorithms to model the nonlinear relationships between market data and other inputs.

## 2.2 Sentiment Analysis

Inclusion of sentiment analysis to predict index movement has shown significant potential leading to a recent increase in papers that have analysed the correlation between public sentiment and market sentiment [8][9][02]. Additionally, the web has gradually become the carrier of much information in our society. Since the popularization of the web, social media websites such as Twitter, Facebook, etc., have become ubiquitous forum for social networking and opinion sharing. By mining and analysing this information, emotional tendencies can be identified, and this information can be used to predict financial movements [11]. Bowell et al. [12] proposed emotional analysis on Tweets in their study and were successfully able to predict the up-down movement of Dow Jones Industrial Average with an accuracy rate of 87.6%.

Market sentiment or “investor sentiment” is another important factor traders consider when making trade decisions as it affects market movements. Kaihui et al. [8] analysed market sentiment using their models and were able to achieve an accuracy rate of 70%. Deng et al. [11] employed a Dirichlet Processes Mixture model to develop a time series of tweet sentiment which was then regressed with the S&P100 index to achieve a prediction accuracy of 68%.

## 2.3 Machine Learning

### 2.3.1 Deep Learning

The recent development of the multi-layer concept in machine learning has allowed researchers to use deep learning neural networks as prediction tools. Largely, convolutional neural networks, deep belief networks and stacked autoencoders approaches of deep learning have been widely used in studies predicting financial instruments’ future trends [12]. For example, [13] employed a combination of the neural tensor network and the deep convolution neural network in their study to predict how events influence the long-term and short-term stock price movements. Furthermore, several works such as [14] and [15] used a deep belief network with conjugate gradient method and restricted Boltzmann machines respectively to predict financial trends. Both papers managed to fine tune their results and achieve a better accuracy rate.

Researchers have also applied Artificial Neural Network (ANN) approaches for time series forecasting of financial data which have failed to provide encouraging results [9]. However, a combination of the 2-Directional 2-Dimensional Principle Component Analysis ( model and Radial Basis Function Neural Network model has produced favourable results. Additionally, [9] proposed a combination of and Deep Neural Networks which improved on the accuracy rate by another 4.8%.

### 2.3.2 Cross Domain Analysis

More recently, some studies have tried probing into the realm of cross domain analysis. As explained by [16] and [17], there is an evident correlation between markets and financial instruments. Investigating these relationships between different markets, financial indicators, financial instruments has shown the potential to yield increasingly favourable results.

A screenshot of a cell phone

Description automatically generated[2] has further investigated how cross domain analyses can be beneficial to financial forecasting. They proposed a cross-domain based deep learning approach. They employ a multi-task learning architecture that uses Recurrent Neural Networks, Inner Domain and Cross Domain Neural Networks to capture time series correlations and then use the information gained to make multi market predictions [2]. Their research has proven that using cross domain analysis can provide superior performance than other basic models. Figure 2.1 illustrates the complexity involved in financial predictions while considering multiple domains.  As explained by [2] financial markets are affected by the interactions between homogeneous markets, i.e. inner domain correlation; the interactions between heterogeneous markets, i.e. cross domain correlation; and the time series correlation, i.e. the correlation between time T-1, T and T+1. Analysing inter-domain correlations resulted in higher prediction accuracy.

Figure 2.1 Correlation Between Multiple Financial Markets and Domains [2]

Another research that has shown significant results is [3]. The paper proposed a cybernetic system that combines the advantages of econometrics and machine learning modules Neural Networks (NN) and Genetic Algorithms (GA). The experiments resulted in the proposed system outperforming benchmark neural network by 35% in predicting the Asia Pacific stock market. Figure 2.2 explains the process used by [2] in his study. The multivariable time series data is firstly fed into the vector autoregression (VAR) analysis, and then the NN determined by the VAR analysis. Lastly, GA is used to deal with the various correlations in data and produce an output [3]. Therefore, multi-market analysis could yield in better insights and substantially improve prediction accuracy.

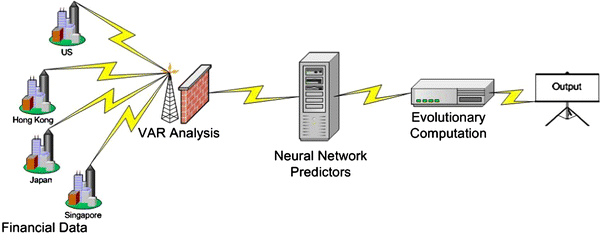


Figure 2.2 System Diagram of Ao's Proposed Cybernetic Mechanism [3]

## 2.4 Time Series Analyses

Several research papers have focused their research on analysing correlations between historic financial data and its ability to predict future returns [16] [17] [18]. Along with traditional time series models discussed before, Support Vector Machines (SVM) have shown significant performance and versatility in the field of time series analyses. Thissen et al used a modified SVM to predict financial data in their research and showed that SVM could perform equally as well as ARMA models [19]. Widow et al and Zbikowski highlighted the versatility of SVM based models [20][21]. [20] showed that SVM models can be adjusted to automatically select suitable lag period for time series data by using multiple kernel learning techniques. This model also showed superior performance as it could adjust for every training set, leading to more variance. [21] highlighted that a weighted kernel could be developed for SVM to adjust the cost function and give more preference to days when high volume of trade took place. This technique was further fine-tuned using a walk-forward approach and feature selection using Fishers method. This research showed that the using weighted kernals in conjunction with feature selection significantly improves on same trading strategy in terms of overall rate of return [21]. Therefore, SVM was highlighted as a possible model for further testing.

## 2.5 Summary

As seen in part 2, several different approaches using linear models, machine learning and statistics have been used in the past to tackle this abstruse problem of financial prediction. However, inadequate research on understanding how the relationships between various financial instruments and market sentiment can be used to predict market movement is observed. Therefore, an approach involving the same will be used in this project. Furthermore, support vector machines have shown significant usefulness in tie series analyses and would be a major focus in this project. The following section will elucidate our approach for this project.

# **METHODOLOGY**

## 3.1 Introduction

Before beginning the project, software development practices and goals of this project must be established. One of the major goals of this project is data collection. Data must be collected from various sources like social media, financial websites, etc and processed for use. This will involve various sub steps such as exploratory data analysis, feature engineering, etc. Subsequently, an iterative process, aiming to forecast financial data, will take place by attempting to improve parameters for the model. Later iterations in our project will call for more sophisticated machine learning models to be used. This section will present a detailed explanation of our proposed approach to tackle this problem.

## 3.2 Programming Language

Python was chosen as the default coding language for the project. Python is well known for its extensive libraries focused on data analysis and therefore, shines out as a prospective choice as the programming language. Moreover, python has a very useful library called “Scrappy” that provides web scrapping tools to the user. Furthermore, python’s syntax is easily readily and least redundant. Hence, python was the most obvious choice.

## 3.3 Software Development Practices

An agile approach to software development is being used in the project. Agile software development is a software development methodology based on an iterative development process. Goals are set forward at the beginning of the project and user stories (description of various use cases/ requirements of the project) are developed. These goals can evolve with time and are assessed at the end of each sprint (one iteration of the process) as new challenges emerge or new requirements are generated. The solution/project grows in an iterative process with high risk high value user stories completed during the early stages of the project. This keeps risk low as the team can look for different approaches to the project at an early stage. The schedule of this project contains the due date of each sprint and the goals we plan to achieve in each sprint.

## 3.4 Data Preparation: Step-by-Step Approach

In order to run our machine learning algorithms, we must collect the right data and engineer it as per our requirements to derive favourable results using available and self-developed forecasting approaches. Thus, the following steps were necessary before we delved into the iterative process of fitting, improving and evaluating machine learning models. Figure 3.1 shows the process of data preparation that we practiced.

Figure 3.1 Process of Data Preparation Involving The Following Steps: Data Collections, Data Processing, Sentiment Analysis, Exploratory Data Analysis, Sentiment Analyses And Feature Engineering

Data Collection –

Web scraping

Data Processing –

Cleaning raw data

Sentiment Analysis

Exploratory Data Analysis

Repeat if necessary

Feature Engineering

### 3.4.1 Data Collection

Data collection is an important step to define the domain in which we aim to apply our forecasting algorithms. The scope of this project was limited to Asia Pacific region and therefore, market data from major financial hubs in Asia was collected to form the domain of this project. The choice of problem statement entailed collection of financial data from multiple financial domains like forex rates, stock indices, futures rates, etc. Since stocks and futures are relatively popular choices of data for financial analyses, forex and indices were chosen for the purpose of this project. Data from Twitter was also collected to act as the domain on which sentimental analysis would be conducted. As observed, data collected can classified into numeric and non-numeric types. Market related data make up numeric time-series data and social media related data like tweets make up the non-numeric or textual time-series data. Numeric data is further differentiated based on source and non-numeric time series data (social media data) is classified based on keywords used to scrape that data. Data is collected using a scrapper for Twitter data and APIs for market data. The following subsections will elaborate on the working of the chosen scraping framework and APIs.

#### **3.4.1.1 Web Scraping Framework: Scrapy**

Market sentiment often plays a key role in understanding trader’s sentiment. To decode market sentiment and find relationships between market trends and market sentiment, non-numeric data was collected to represent human sentiment. As a popular platform to voice one’s opinion, Twitter was chosen as the main source of market sentiment. Since every website has data presented in a unique way, it is essential to develop a custom scrapper for the website. We use “scrapy”, an open source web scraping framework developed in python, to develop a scrapper for Twitter. Scrapy allows for developing a program highly focused on code reusability. It has the ability to crawl many web pages in a relatively short amount of time and therefore is preferred over directly using Twitter’s APIs for extracting tweets. Figure 3.2 gives an overview of how a web application written using scrapy extracts data from the web. The application sends a request to the web browser for the requested data. The browser responds with the webpage from which data can be extracted and processed.

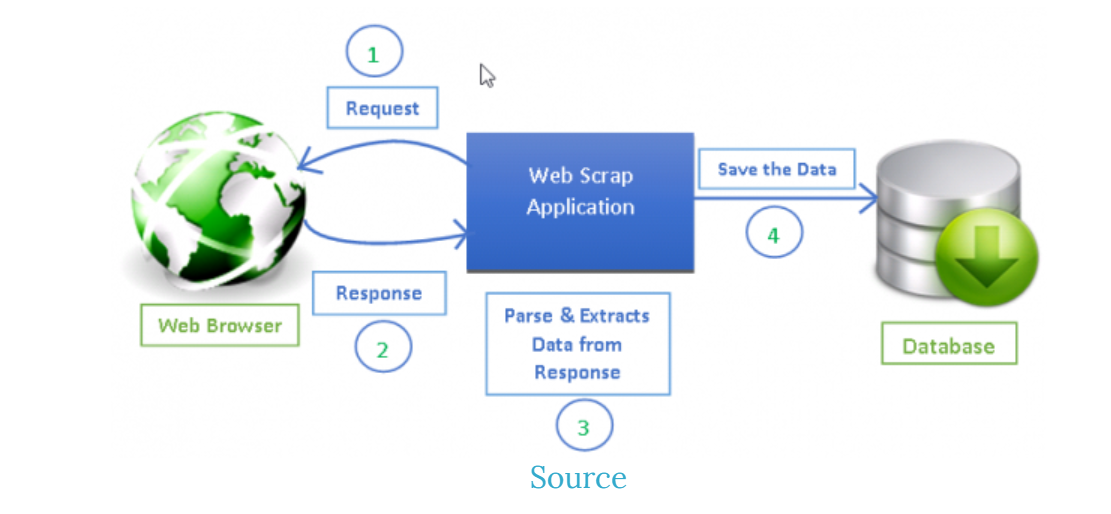


Figure 3.2 Web Scrapping Process Using a Web Scraping Application

#### **3.4.1.2 Market Data API: InvestPy**

Investpy is a python project to retrieve financial data. A simple call to this API with relevant parameters, namely, stock name, country and from and to do returns basic information related to stock like open and close price, daily high low, etc. By using this API, we can retrieve historical data for stocks, warrants, forex and indices.

For the purpose of this project, 2 forms of numeric data were selected to facilitate discussions on cross domain correlations in markets. Firstly, values of market indices of markets in APAC were chosen. Market index data of the primary index of 18 countries in APAC were collected. In addition to market data from Asia and Australasia, Market indices of USA were chosen as a base reference. Open High Low Close Volume (OHCLV) data for each trading day were collected for all of the target markets using InvestPy.

Secondly, currency exchange (“forex”) rates for currencies native to the countries in scope were also collected. A currency’s exchange rate with respect to the United States Dollar constituted this part of numeric data. Forex rates for all trading days for individual markets were collected in the form of XYZ/USD where XYZ represented the market’s home currency. This was obtained in the Open High Low Close format (OHLC).

### 3.4.2 Data Processing

Before this data collected can be used, it must be processed which makes information extraction easier. Both, textual and numeric data are handled separately to extract the most information out of them.

Due to the unavailability of perfect financial data, the numeric data was cleaned to deal with missing value. Moreover, consistency was established between different markets as each country had their own trading laws. Missing data was handled primarily in two ways, interpolation, or removal of data. If the amount of missing values were quite small, then that entry was discarded from the database. Once such data were removed, all markets were aligned by taking US trading days as the standard. For example, countries where trading took places between Tuesday and Saturday, their trading days were adjusted to match the US. Following this, any missing data was handled through interpolation. The data was also scaled to identify outliers. Any point identified an outlier was replaced with a value equal to two times the standard deviation. Data was then be canonicalized to achieve consistency. Canonicalized data is imperative to effective functioning of our models.

To integrate textual data along with the numeric data, it too was canonicalized into numeric format. To achieve this normality, sentiment analysis was performed on the tweets collected using the scrapper for twitter. By using an open source software such as Bidirectional Encoder Representations from Transformers (BERT), process of sentiment analyses was streamlined and efficient enabling the processing of tweets to obtain their sentiment scores. These scores can be interpreted as probabilities indicating whether the text is positive or negative. The results obtained after applying these algorithms can then be aggregated over the desired time period for each category (keyword) of data. These sentiment scores represented market sentiment. Therefore, mean sentiment was calculated for each day and matched with the date index of numeric data. market sentiment of the weekend was grouped with Friday to act as input to predict market movement on Monday. This eases interpretations of trends as all data is represented numerically and consistently.

### 3.4.4 Exploratory Data Analysis

The numeric data obtained is then subjected to data analysis. This analysis consists of several parts targeting the different aspects of the time series and non-time series elements of the data. This part of the project holds greater significance to the project. Correct analysis of data can glean insights on the potential machine learning models suitable for the data. Understanding and transforming data is imperative to improve predictive capabilities of any machine learning models

Exploratory Data Analyses (EDA) was part of the project which investigated the data to identify patterns that can be exploited to ease the process of machine learning. It also helped us conduct initial analyses of the raw data. This enabled us to draw relevant conclusions and derive relevant features that could act as potential parameters for the learning models. For time series data, properties such as seasonality and trend were identified through seasonal decomposition to exploit during feature engineering. The data was also tested for stationarity which was handled later. Feature importance analysis therefore helps identify the most relevant features which can facilitate assigning weights during the forecasting process if required.

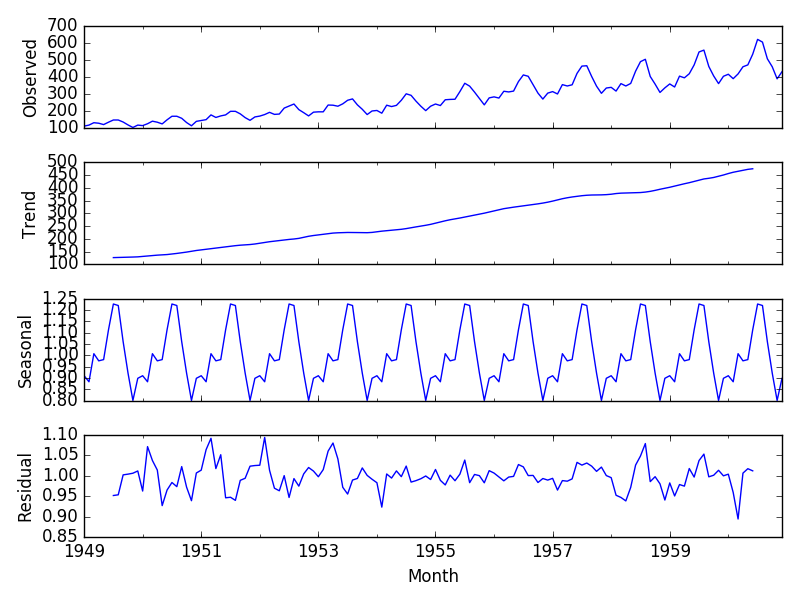


Figure 3.3 Seasonal Decomposition of Close Values

Figure 3.3 shows the seasonal decomposition of time series data clearly showcasing the seasonality of the data. Such seasonality can easily be exploited in machine learning where setting a seasonal component can achieve better results.

### 3.4.5 Feature Engineering

Based in the results obtained from EDA, feature engineering was performed to manipulate existing data to obtain relevant features and greater insights. there are steps in feature engineering based on the machine learning models chosen. Naïve machine learning models usually do not implicitly evaluate the time series nature of the data. Therefore, it is important to provide it with a few values from the past to allow previous records to be used as features too. This was achieved by both, directly including the raw values of time period (t-1) in row (t) and by calculating the rates of return over the periods thereby further simplifying the data. Furthermore, features like moving average could also be beneficial to such machine learning models. Therefore, several return values pertaining to intraday and inter day were calculated to facilitate classification. Moving average was derived for the purpose of back testing model performance and testing model accuracy qualitatively.

## 3.5 Iterative Algorithm Development

After having prepared the data for forecasting, we begin to apply it using forecasting functions in an iterative process while determining the best parameters for each of the different models and scenarios. Two kinds of models: time series regression models, and naive classification and regression models were chosen for this iterative process.

The iterative process involved splitting the data into training and testing sets, selecting a set of parameters, training a model, making predictions and evaluating the performance of the model as explained in the figure below. This whole process can be repeated several times to also optimize parameter set for an induvial model. Figure 3.4 shows the basic algorithm development process.

Figure 3.4 Algorithm Development Lifecycle

### 3.5.1 Test Train Split

Size of the training data set can heavily influence a model’s performance. Larger training datasets have a higher chance to truly represent the data and have enough samples to highlight the patters that can enhance prediction accuracy. An effective train test split needed to be identified that would contain enough information to predict the test data accurately. For naïve methods, random sampling is an explore worthy option as these models as these models look for relationships between the data and output and doesn’t analyse the time series aspect of the data. A cross validation approach is feasible here to split the data. It has an added advantage of improving the prediction accuracy as the model penalises itself for the loss incurred at each iteration of cross validation. For methods that implicitly factor in time series aspect of the data to produce predictions, such a split of training and testing data is not feasible. Auto Regression Moving Average (ARMA), is such a model. Here, creating a single train test split is advantageous as we can take advantage of the moving average time series analyses along with auto regression capabilities to gain meaningful predictions.

### 3.5.2 Parameter Selection

The parameter selection step is usually two folds. There are two kinds of parameters that need tuning in any machine learning parameters. One set of parameters is the parameters learned during the machine learning process. As explained in section 3.5.1, K-fold cross validation can be used for tuning of these parameters. Due to the nature of sample data, a different train test split can often lead to different performance. This can be attributed to the variance of the dataset. By using K-fold cross validation, these internal parameters can be tuned to effectively represent the variance of the entire data set. This can lead to an increase in testing accuracy.

Another set of parameters that need tuning in a machine learning problem are the hyperparameters supplied to the model. Kernals in SVM, K neighbours in KNN algorithms, etc are examples of these parameters. In order to tune these parameters for naïve models, a varying parameter set can be supplied to a model and compared to the base model with only default values. Multithreading is a popular technique to improve performance and test several models simultaneously without incurring overhead associate with multiprocessing. For more sophisticated models, parameters can be tweaked manually. Best performing hyperparameters are taken for following steps of training and testing.

### 3.5.3 Training and testing

Machine learning by virtue aims at identifying the nuances in the feature set in order to create reasonable assumptions satisfying the hidden function dictating the behaviour of the target variable. The aim of all machine learning algorithms is to find the weights associated with each training feature that best resemble the hidden function f(x).

All training and testing data is split into feature variables and target variables. Feature variables included OLHC values of the target market along with several features engineered during the previous steps. This included variables like intraday return, open return, moving averages, among several others. This project focused on targeting the day’s. close for regression models and direction of return on close for classification models. Classification approaches were also used to predict bins of the price range. Models including Ordinary Least Squares, Ridge, Lasso, ElasticNet, KNN, Naïve Bayes, Support Vector Machine (SVM) and Random Forest were deployed during the early iterations of model experimenting. The following iterations focused on using time series modelling using statistical models. Models such as Auto Regressive Integrated Moving Average (ARIMA), Facebook’s Prophet and variations of ARIMA were deployed in this phase. Finally, neural networks were experimented with to understand their added benefits to this problem statement.

### 3.5.4 Evaluation Metric and Techniques

Model evaluation is key to understand the progress of the project. The results across models was consolidated to clearly identify models which could fit the data. Since target metrics for this project were of two kinds, continuous and discrete, different metrics were considered for different classes of models dealing with a particular kind of data. For regression problems, Mean Squared Error (MSE), Root Mean Square Error (RMSE), R-Squared Score (R2), were used. MSE simply represents the sum of the difference between true value and predicted value squared, divided by the number of values. RMSE is the root of MSE. Finally, R2 represents the goodness of fit. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively Following image shows the formulation of these metrics. Here ŷi represents the ith predicted value, yi represents the true ith value and ȳ represents the mean of the data.

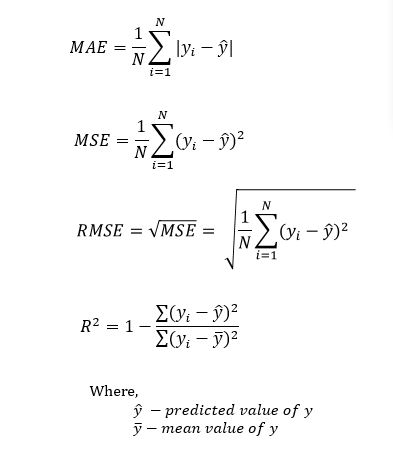


Figure 3.5 Evaluation Metrics for Regression

For classification models, metrics like F1 score, accuracy, true positive rate and true negative rate were used. The F1 score is a good measure of it identifies a balance between precision and recall. Precision is defined as the ratio of true positives and predicted positives and recall can be defined as the ratio of true positive and actual positive. Their formulations are described in the figure below. In practice, not all errors are created equal, with some errors leading to more loss. F1 score is a good base metrics to identify which classification models are performing better in general. Performance of classification models were further analysed using back testing strategies. Back testing is explained in more detail in the following section.

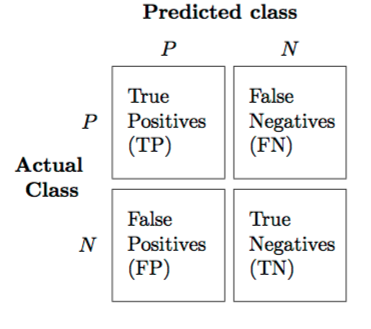
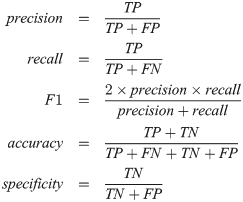


Figure 3.6 Evaluation Metrics for Classification

### 3.5.5 Back testing

A model’s predictions can be provided to a back-testing strategy to test the quality of predictions. A simple trading strategy could be implemented to take advantage of the model’s prediction. Equity return after successfully backtesting a strategy would help his qualitatively analyse the model’s performance. As mentioned in section 3.5.4, not all errors are created equal. Correctly identifying major trends in the market could lead to higher returns and therefore would suggest a higher performance of the model qualitatively. Therefore, back-testing was also chosen as an additional feature in model evaluation. For the purpose of this project, two kinds of backtesting strategies were developed 1) a simple moving average strategy over prediction results and 2) an intraday trading strategy based on predictions.

A simple moving average strategy has been proposed as a high performing backtesting strategy by several papers [18][21]. The project developed a variant of a simple moving average strategy. Instead of computing moving averages over market value, the strategy developed computes a moving average of the predictions. Since the aim of backtesting is to analyse the quality of predictions, it was hypothesised that if the general trend predicted by the model was in line with the actual market trend, the strategy would perform well and the quality if predictions would be deemed high. Therefore, moving averages over a period of 15 and 60 days initially. This was further optimized using multithreading to find optimal window periods for rolling averages and rolling sum of predictions. Figure 3.7 shows the process of calculating moving averages and the strategy on buy and sell.

Predictions for the testing period

Take rolling average of n days.

|  |  |
| --- | --- |
| Slow Moving Average | Fast Moving Average |
| Sman1(t) | Fman2(t) |

n1 > n2

Buy @ t+1 close

If

sma > fma &

pred [t+1] == -1

Sell @ t+1 close

If

sma < fma &

pred [t+1] == 1

Figure 3.7 Slow Moving Average Backtesting Strategy

The second strategy developed focuses on the analysing the quality of predictions and whether it can home in on major downfalls and shocks and help save losses in equity. The trading strategy was based on the strategies of “short selling” and “long buying”. In short selling, the trader speculates that the price of the share will fall in the near future and therefore calls in for a sell order to maximize profits and buy the sold shares back at a cheaper price. The long buy technique is just the opposite of short selling where the expected growth in the market is upwards rather than downwards. Here shares are bought at a cheaper price with hopes to sell at a higher price down the line. This strategy was adopted on an intraday bases, i.e., trades were executed on open and close value for the day.

Predictions for the testing period

Look at prediction of time t+1

Sell @ t+1 Open

Buy @ t+1 Close

Buy @ t+1 Openn

Sell @ t+1 Close

Pred [t+1] == 1?

True

False

## 3.6 Optimization

The algorithm development was further optimized using various techniques. Optimization step takes advantage of multithreading in python to simultaneously build and compare models with millions of combinations of hyperparameters. This makes the process of finetuning parameters for each market simpler and easier. Each thread takes a set of parameters which are used to build a model and the predictions from each model is compared to the others to determine the best parameters to increase accuracy. Another way to optimize a model’s performance is to fine tune the input size in order to avoid overfitting and underfitting. Feature importance was also analysed to determine feature sets that account for high proportion of variance in the dataset. The following subsections will discuss regarding the implantation of three types of optimizations techniques employed in this project in more detail.

### 3.6.1 Walk Forward Classification

The nature of time series data dictates that the data needs to be sampled and trained sequentially. To avoid loss incurred due to random sampling of data, train-test split was done in a linear manner. However, a single train-test of 80-20 as explained in previous sections would lead to overfitting [Żbikowski, 2015 #21]. The correlation between historic data and predicted value decreases exponentially with time. Therefore, to get optimal results, a sliding window walk forward testing was developed. It was hypothesised that market sentiment highly influences the market movement and hence a suitable window of historic time period directly before the test value would include relevant information to identify the trend in market sentiment. As shown in fig below, (t-window\_size) to (t) indexed features are taken from the feature set to predict (t+1). The window is the moved forward by one day to predict (t+2).

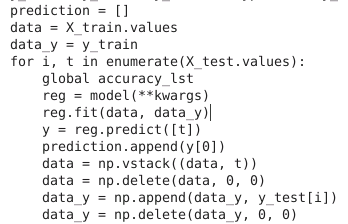


Figure 3.8 Algorithm for Walk Forward Classification

This step has the potential to be repeated with several window sizes to determine the window size that best explains the variance in data. Moreover, similar to one vs rest classification strategy of SVM, a mode of predicted values can be taken across all different window periods to get an average of the predicted results.

### 3.6.2 Multithreading

Multithreading technique is a simple way to build thousands of models simultaneously. Each model can take a unique set of parameters. These parameters dictate the way the model learns from the data and the loss it incurs at each step. In order to optimize weights associated with each input feature, different set of parameters can be tested. Accuracy of the performance of each model is assessed in order to identify a set of parameters that help the model learn and find the optimum decision boundary. The accuracy of these models is assessed based on F1 score and accuracy scores. These selected parameters are also assessed in backtesting and compared against each other.

### 3.6.3 Feature Selection

Often in a high dimensional feature set, there remain several redundant features. These redundant features do not effectively contribute to the model training. Hence, there is a need to extract the most important and the most relevant features from the dataset in order to get the most effective predictive modelling performance. This step in optimization pipeline refers to feature selection. Feature selection enables the machine learning algorithm to train faster. Moreover, it reduces the complexity of a model and makes it easier to interpret and improves the accuracy of a model if the right subset of features is chosen. Therefore, feature selection was chosen to further fine tune models and their performance. Principal Component Analyses (PCA) was chosen to extract features and reduce dimensionality. PCA works by calculating the covariance matrix is calculated to represent how each variable correlated with another, determining the Eigenvectors which represent the direction in which the data is dispersed, and calculate the relative importance of the Eigenvectors. PCA combines these metrics and drops the unimportance Eigenvectors such that most of the variance in input data is represented. This can be done by calculating the variance explained by each feature and adding features until a desired threshold is achieved.

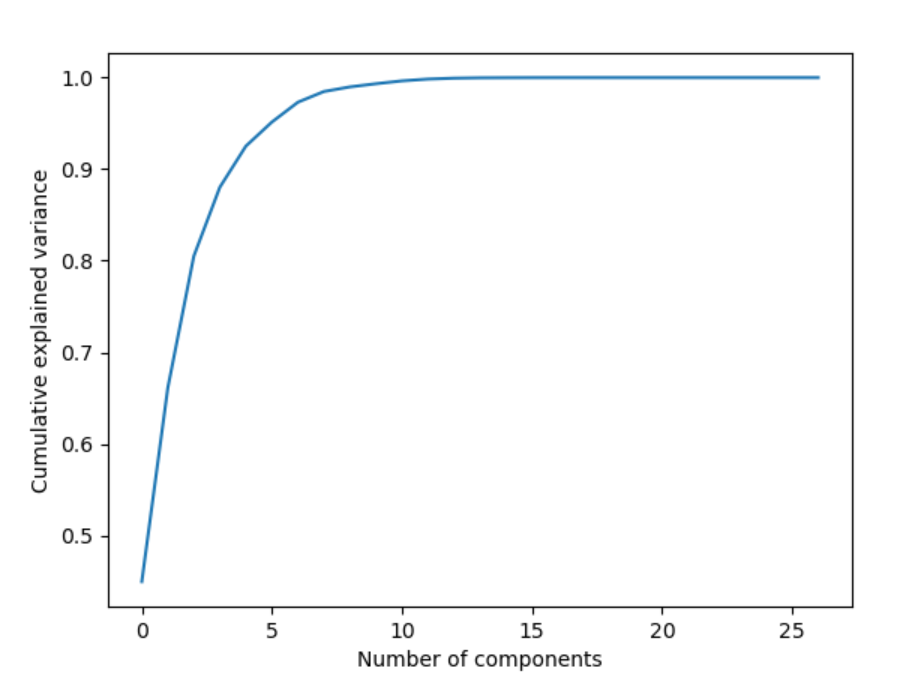


Figure 3.9 PCA Analysis of BDT. Image Shows That Majority of The Variance is Explained by 8 Components for BDT.

## 3.7 Summary

Section 3 elaborated on the approach adopted for this project. Literature review highlighted the importance of a step by step approach to a project of this nature. Therefore, an iterative software development process is chosen to ensure steady progress in project development. Data preparation is the one of the key components of this project and therefore a major portion of this project will be dedicated to collecting the right data and inferring trends from it before prediction algorithms can be employed. We aim to use algorithms with increasing complexity and devise potential ways our inference from the data can be used to make better predictions as the project progresses. Once satisfactory results are obtained, we aim to begin development of our own algorithm which could increase on accuracy rates of our predictions.

# **PROGRESS, EXPERIMENTS AND RESULTS**

The team employed an agile software development process with biweekly sprints. We focused on reviewing related works for most of September 2019 to thoroughly understand the challenges faced by others and possible approaches that we can take in order to determine the gaps in research on this topic. A project website was set up to display progress, set up a basic web framework to deal with data collection and processing and exhaustive testing was carried out on varies machine learning and time series models. The website of this project can be found at: [*https://i.cs.hku.hk/fyp/2019/fyp19020/*](https://i.cs.hku.hk/fyp/2019/fyp19020/)

Scripts were written to facilitate the progress of the project. A set of scripts would focus on a problem and would come together to forma pipeline. Scripts written for data collection would primarily form one pipeline and were set up to run periodically to collect data. For experimentation, scripts written for each experiment would iterate over the data for chosen markets and currency and would act as input for provided models. An inclusion exclusion strategy was also carried out to understand the effects of feature reduction, sentiment scores, inter domain and intra domain analysis. While evaluating available models, experiments were carried out using endogenous lagged variables of the currency/index first. Models which provided adequate performance were then probed further with the cross-domain approach and further optimized using walk forward techniques and hyper parameter selection.

**4.1 Data Collection**

Two different kinds of data were collected for this project as mentioned in section 3.

### 4.1.1 Numerical data

For numerical data, investpy was identified as a source to extract relevant market data. The data collection process involved installing the relevant packages and calling the API to return financial data for required period. Figure 4.1 shows the result obtained by running the following lines of code:

**df = investpy.get\_index\_recent\_data(index='Nasdaq 100', country='united states')**

**print(df.head())**

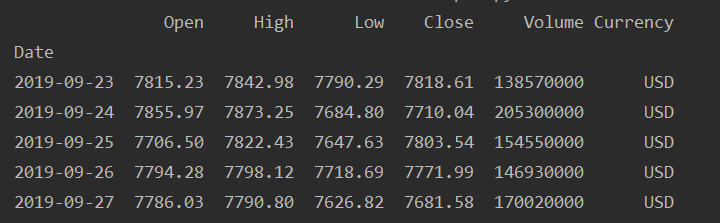


Figure 4.1 Most Recent Data Available for NASDAQ 100

The flexibility offered by the API allowed us to select between different time periods from the inception of the preselected markets until the current date to obtain requested market data. This data was made easy to or kith by storing it in text files which made pipeline for parsing data smoother. Pandas package was used to iterate over the pairs of currencies stored in the text file and all relevant information was consolidated into a single comma separated file for both forex and indices. Owing to the small size of the data, the need to maintain a server to store it was made redundant as it would lead to overhead costs.

### 4.1.2 Textual data

For the purpose of this project, Twitter was chosen as the source of textual data. Although, several APIs exist that allow users to request historical data, Twitter imposed a restriction on the number of tweets that can be retrieved and disallowed retrieval of tweets through an API that were older than six months. These challenges were overcome by employing scrapy to create a web scraping tool as explained in section 3.4.1.1. The working of scrapy was penned down in the same section. A web scraper developed by using a scrapy further provided an added advantage. The queries could further be narrowed down by providing keywords and limiting the results to tweets containing the same. The scope of the project dictated that the keywords be limited such that the tweets represented market sentiment of the region. Therefore, keywords used while scraping included country name, financial capitals and names of political leaders. Posts from users containing specified keywords were scraped and consolidated in JavaScript Oriented Notation (JSON) files on the server.

The virtual machine access provided by the University of Hong Kong’s Computer Science Department enabled the team to set up scripts to run on a periodic basis to query data, transform it, update relevant queries and move collected data to storage directory. Several scripts were built as a pipeline to achieve this goal. A file with consolidated queries was kept facilitating the pipeline. The pipelined involved scripts, each with a specific goal as explained below:

1. Start a background process to scrape twitter using the first incomplete query from the file.
2. Check if a process is complete and update the relevant query in the file.
3. Check for missing dates and update query file with a new query to search for data on missing dates.
4. Convert collected data to a JSON file. A copy of this file with only text was created as a comma separated values file (CSV) for use in the following steps of data processing.

The result was then stored in an appropriate folder to be later used in Sentiment Analysis. Log files were maintained and used to track of data collection progress and errors. Figure 4.2 shows a JSON object from the file produced by Script 4.

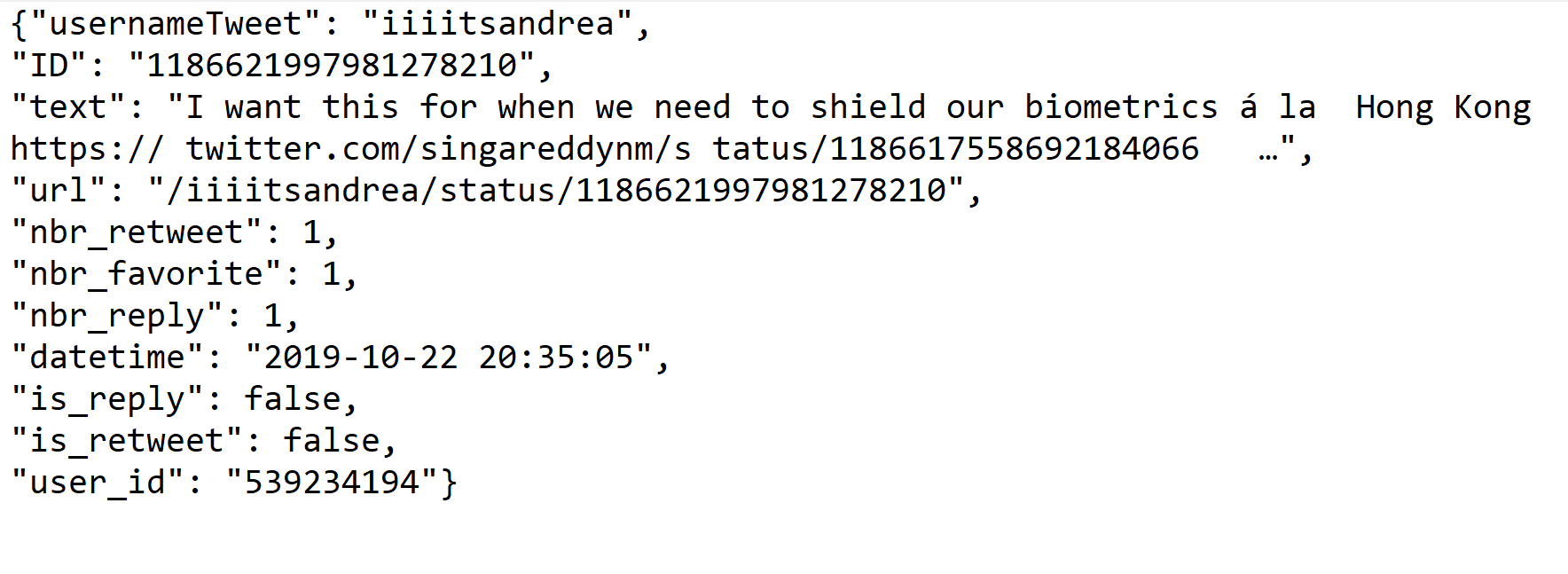


Figure 4.2 A JSON Object Containing Information About the Tweet Retrieved from Twitter

## 4.2 Data Processing

The data collected in the previous step was subject to different kinds of process and feature extraction. Textual data was subject to sentiment analysis which aimed at deriving an important feature representing market sentiment. Numeric data was subject to data cleaning and pre-processing before financial indicators could be derived from them. The following subsections elaborate on the results from these steps.

### 4.2.1 Sentiment Analysis

Bidirectional Encoder Representations from Transformers (BERT) was the basis of developing a sentiment model in this project. BERT is a publicly available model which works by bidirectionally parsing a sentence, thus building contextual relations for the words forming a sentence. Moreover, this is viewed as an advantage over methods that parse sentences unidirectionally, leading to an overall higher accuracy.

Multilingual Cased model provided by Google was used to train a BERT model for the project. Masked Learning Method (Masked LM) and Nest Sentence Prediction (NSP) are used in combination to train the model and reduce loss caused by error. The model aims to predict scores for certain words which are masked in Masked LM by analysing the rest of the words forming a sentence. In NSP, BERT tries to predict the sentiment of the following sentence by looking at the sentences preceding it. A combination learning method. BERT outputs a pair of probabilities indicating the sentiment of the tweet. These probabilities were then grouped daily and varies statistics were derived from them to mimic market sentiment. Figure 4.3 shows the various features engineered from sentiment results.

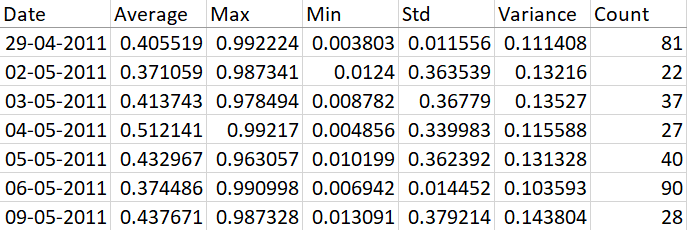


Figure 4.3 Statistically Derived Features from Sentiment Analysis

### 4.2.2 Processing Numerical Data

Due to the varying nature of bank holidays coupled with recent inception of some markets, data gathered was non-uniform and had several missing values. Therefore, first step in processing numeric data involved lining up the working days of all the market. For this purpose, United States of America was chosen as base market and therefore all working days were adjusted to a Monday to Friday work week. Missing data was then dealt by applying interpolation. Each time series was interpolated for the missing values since the inception of the market using component of the time as a factor to calculate the magnitude by which the values would have changed for the missing time periods.

Figure 3.3 in Section 3.4.4 highlighted the non-stationary nature of raw data collected. Stationarity means that the statistical properties of the process do not change over time. Since no regular trend was observed in raw data, was converted into returns. Returns are calculated by analyzing the movement of an indicator daily. The results from an Augmented Dickey-Fuller (ADF) test confirmed that the raw values were not stationary. Figure 4.4 shows the seasonal decomposition of Hang Send Index returns. It is observed that returns are highly seasonal and lack a general trend.

A screenshot of a cell phone

Description automatically generated

Figure 4.4 Seasonal Decomposition of Return on Hang Seng Index Close

Market indices and forexes use an independent scale and are therefore non comparable. To facilitate cross domain analyses, it was essential to maintain comparable features across the selected markets. Additional features were calculated to achieve this. Figure 4.5 shows the various returns and moving averages derived and their statistical analysis done using the Augmented Dickey-Fuller test (ADF). The results of the test indicate that the derived features are stationary due to a p value of less than 0.05 which is often considered significant. These values therefore acted as input features for the future model learning processes.

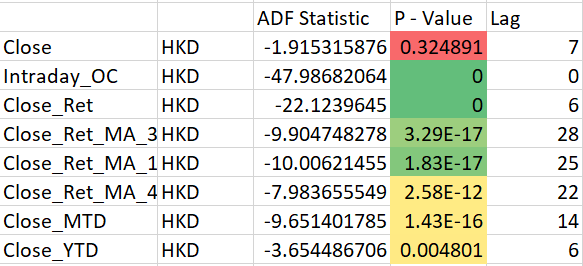


Figure 4.5 ADF Test Results Highlighting Stationarity of Derived Features.

Correlation matrices were also created to help visualize the correlations between different markets. It was used to identify the intra-domain and inter domain pairs which show significant correlation which could be exploited in machine learning. Figure 4.6 shows the correlation matrix for all currency pairs. From the imagine, multiple significant correlation pairs are identified and testing in later processes. (MNT, LKR) is an example of such pair.

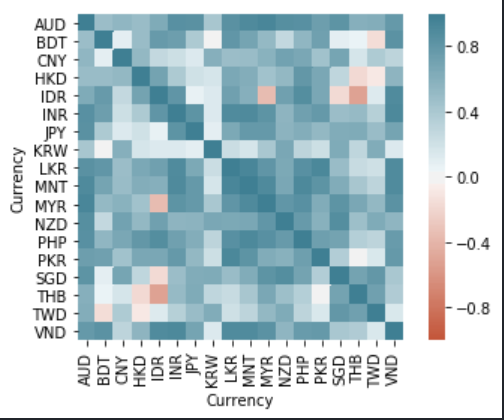


Figure 4.6 Correlation Matrix for Currencies for Time Period 2010-2020

Preliminary data analyses suggested correlations between several pairs of data. To narrow down the scope of this project and in order to provide valuable insights, less researched markets were chosen for the course of this project. Moreover, highly correlated intra domain and inter domain data was identified to act as input in cross domain analyses. Therefore, developing countries were chosen for the project. An interesting observation to be made from the table below is how the chosen markets were correlated with other developed markets. For example, it can be seen that Karachi 100 was highly correlated with Nikkei 225 of Tokyo. This relationship mimicked the real world political and economic interdependence.

Table 1 List of Forex and Indices Chosen Along with Correlated Markets

|  |  |  |
| --- | --- | --- |
| Market Type | Chosen Market | Correlated Markets |
| FOREX | **BDT** | **VND** |
| **IDX Composite** |
| **MNT** | **LKR** |
| **NZX MidCap** |
| INDEX | **Karachi 100** | **INR** |
| **Nikkei 225** |
| **CSE All-Share** | **IDR** |
| **MNE Top 20** |

## 4.3 Machine Learning Experiments

The next step as explained in section 3 was experimentation. Experimentation began with data gathered in previous steps as input. The following subsections elaborate on various machine learning approaches adopted and compares their results.

### 4.3.1 Regression Models

Initial exploration and experimentation focused on regression models. The models were applied on the collected data with close value as target variable. Figure 4.7 and 4.8 shows the results generated from running linear regression on BDT. The model’s predictions suggest a trend. The predictions followed the lagged movements of the market and did not offer any significant insights. Similarity of behavior across all naive regression models prompted the research into a suitable target variable. This research deemed Close Return as a favorable target variable.

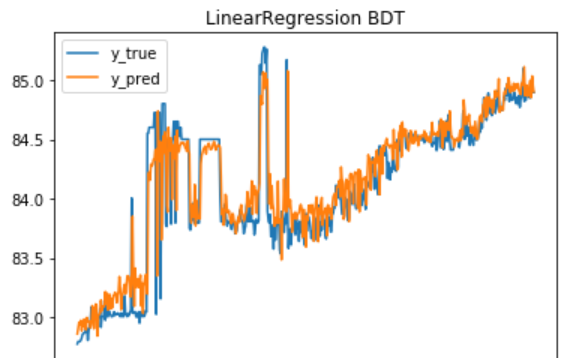
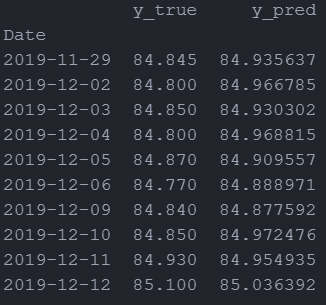
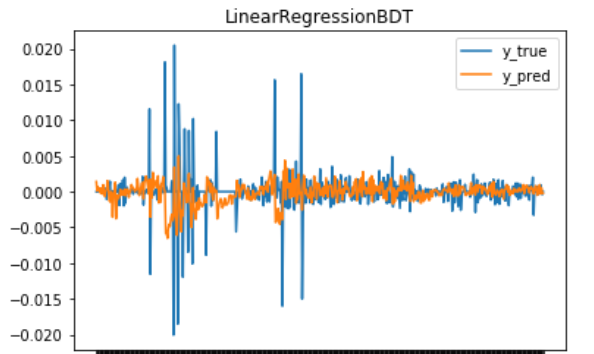


Figure 4.7 Predicted Raw Values by Linear Regression Model

Figure 4.8 Linear Regression Results on Raw Values of BDT

On shifting to predicting returns using regression models, the accuracy scores obtained were insignificantly small and almost negligible. The R2 score of regression models fell but this fall was attributed to the small values of returns. However, as seen in Figure 4.8, one can see that the regression model doesn’t get offset by few large changes in target values and sticks more to the mean. This also further solidifies our claim that returns calculated are trend stationary, i.e., the series converges to the mean and doesn’t get affected by shocks. However, regardless of using this target variable, the models still didn’t give favorable R2 and MSE values.

Figure 4.9 Regression Graph for BDT Close Return

### 4.3.2 Time Series Models

AutoArima framework enables users to conduct a stepwise selection of parameters using the ADF test. Hence, AutoArima was chosen for experimentation purposes. Furthermore, a simple multithreading strategy was used to widen the hyperparameter and configuration set. In this way, AutoArima framework enabled experimentation for ARIMA and its variant SARIMAX by varying parameters like seasonality and exogenous variables.

#### **4.3.2.1 Experimentation**

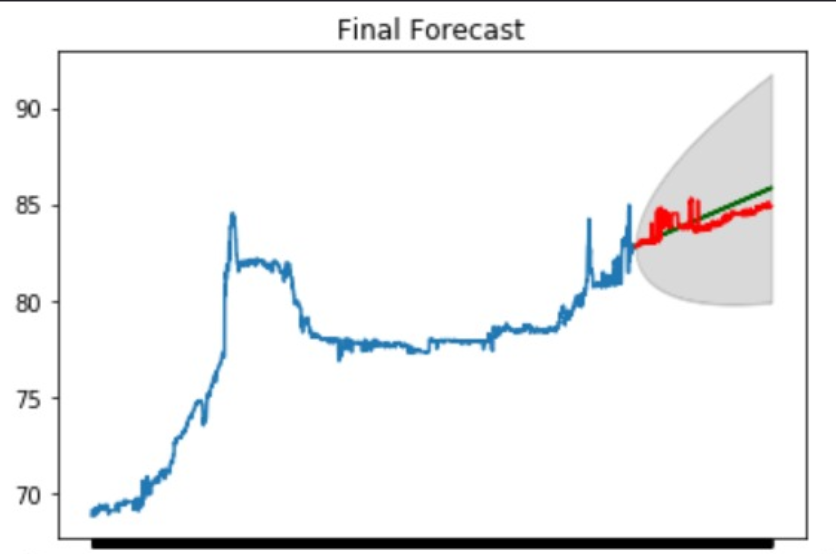
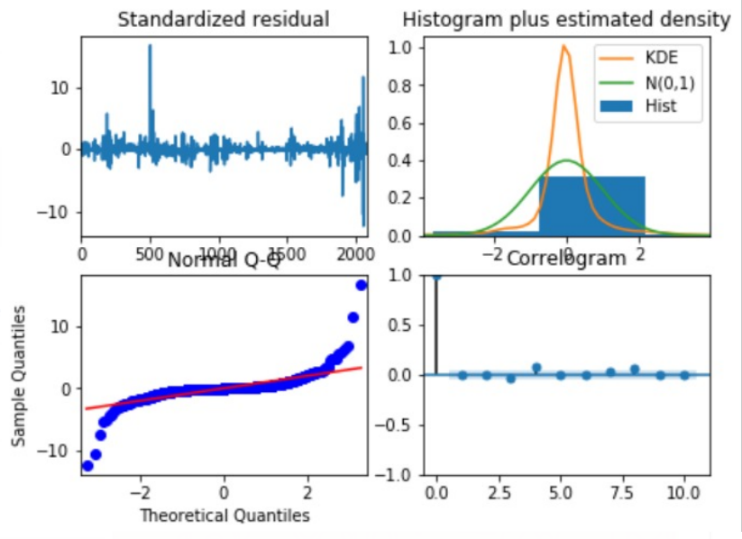
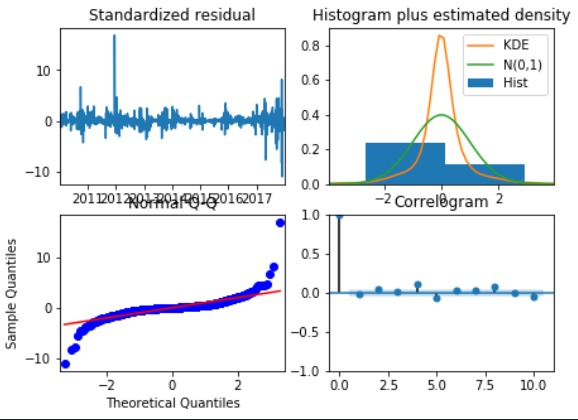


Figure 4.10 ARIMA Diagnostics

Figure 4.11 ARIMA Regression Results Using Raw Values of BDT

Initial experimentation using raw values on ARIMA suggested an improved performance over classical regression models. It was hypothesized that since ARIMA incorporates trends in the time series data, it displayed this improvement. This can be seen in Figure 4.11 where predicted value (green) moves in the direction if the market (red). However, Arima failed to account for magnitude for changes and shocks in market. Moreover, an analysis of the model diagnostics suggest that the data fit was not optimal. The raw values consisted of several shocks which confused the model and are the outliers in Q-Q plots. The KDE curve also does not fit the normal distribution well as well. The model simply follows the recent average trend of an upward direction.

The Next experiment entailed using returns values similar to the approach for regression models. Derived lagged features were supplied as inputs to ARIMA and this showed a relatively better performance. Figure 4.12 shows the results using derived features. The predicted values (green) follow the trend of true values (red) more closely here. Moreover, the fit diagnostics suggest a fewer number of outliers and a better fit on data as seen in figure 4.13. The better performance can be attributed to the fact that return have a highly seasonal character and follow a flatter trend as shown in Figure 4.4. Therefore, ARIMA was more successful in following the trend and volatility of the target variable.



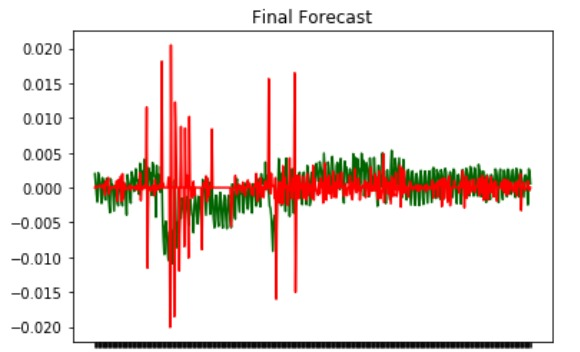


Figure 4.12 ARIMA Returns Forecast for BDT

Figure 4.13 ARIMA Fit Diagnostics Using Returns

Finally experiments entailing Facebook’s Prophet were conducted. It was hypothesized that Prophet would have higher accuracy as it combined knowledge from a combination of statistical models for time series analysis. Figure 4.15 showed that, on raw values, the Prophet model simply followed the generally, exaggerated upward trend of the training data. Using returns, Prophet followed the trend in returns more closely but predicted negative returns more often. The results for returns are shown in figure 4.14.

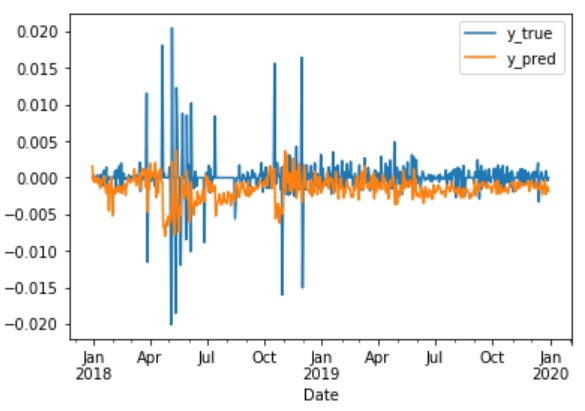
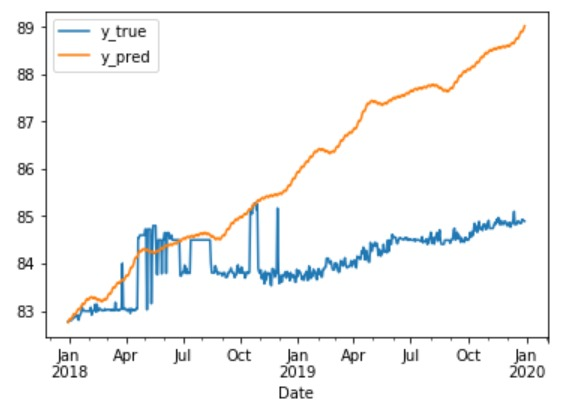


Figure 4.14 Prophet Regression Results Using Raw Values for BDT

Figure 4.15 Prophet Regression Results Using BDT Returns

#### **4.3.2.2 Results**

MSE was used as an evaluation metrics for ARIMA and Prophet. Table 1 shows the MSE scores of the two models using different input parameters. Overall performance of ARIMA surpassed that of Prophet. ARIMA account for moving averages while forecasting and thus could predict results closer to true value. Prophet on the other hand accounts for trend while forecasting and thus performed better in predicting returns where the trend was relatively flat. However, since the R2 scores of neither of these methods were acceptable, these methods were not moved developed on further.

Table 2 MSE Scores for ARIMA And Prophet

|  |  |  |  |
| --- | --- | --- | --- |
| MSE Values | Exogenous features | ARIMA | Prophet |
| Raw values | True | 0.55 | 2.59 |
| False | 0.55 | 5.40 |
| Returns | True | 1.50e-05 | 1.05e-05 |
| False | 1.50e-05 | 8.37e-06 |

### 4.3. Classification Analyses

Following the experimentation of regression models, the team moved to analyzing the performance of machine learning models in classification problems. For the purpose of K-Bins and Binary classification, derived features were used as indicators as most markets followed a unidirectional trend where the prices would fall or rise continuously for a period.

#### **4.3.3.1 Experimentation**

Experiments were performed using by classifying returns into different number of bins to mimic a K-bins classifier and a Binary classifier. Initial exploratory data analyses suggested that several values for returns were far smaller than 1. Therefore, a small number of bins was chosen for K-bins classifier as splitting the data into too many groups would spread our resources too thin and may lead to overfitting of data. This would create a highly biased model with low variance which is unsuitable for future classification. Thus, target variable in available data was divided into four bins using quantiles. Features were not subject to this division and only witnessed scaling to speed up model fitting process. The K-Bins classifier ran over all classification models available under scikit-learn library.

Following the experimentation on K-bins classifier, a binary classifier was developed that would classify results in a positive or negative direction with aims to develop a trading strategy developed using our predictions. The returns were classified into positive and negative classes depending on whether the return was greater than or less than zero. Similar to the last experiment, this binary classifier was used to classify input before fitting the models. For the purpose of both the experiments, out of sample testing was done by splitting the data into an 80-20 test-train split. Top performing models were taken forward into the optimization phase of the project.

#### **4.3.2.2 Results**

Of all the classification models availably in Scikit-learn library, Random Forrest showed the best performance for K-bins classifier and its results are shown in figure 4.16 and 4.17. The confusion matrix shows a better performance than regression model with an accuracy rate of 39%. The low recall and precision are attributed to the high degree of misclassification, especially for class. On further investigation, it was found that that this was due to the relatively close boundaries between class 1 and 2. Moreover, we had noticed during EDA that the last 20% of days had a few major shocks which could not have been predicted accurately by simply looking at historic data.

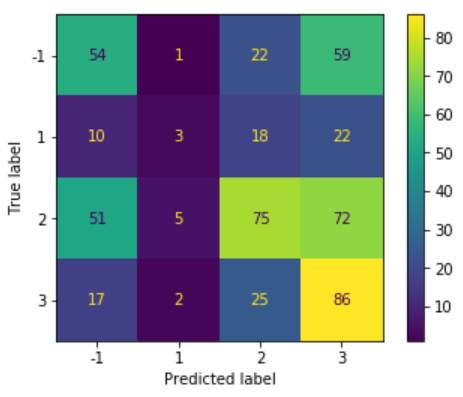
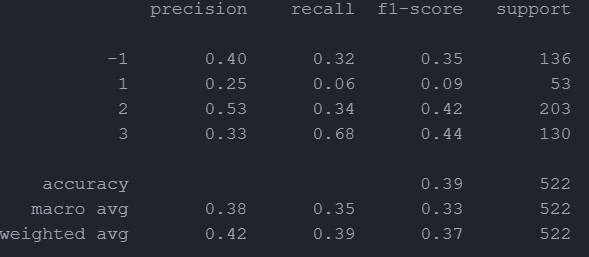


Figure 4.16 Classification Report for random Forest

Figure 4.17 Confusion Matrix for Random Forrest Classification on BDT

Binary classification performed better than K-bins Classifier on average with accuracies reaching as high as 62%. For markets with a relatively flat structure like HKD/USD forex, accuracy was only 52% suggesting no such correlations between derived metrics and the predicted variables. However, for markets like BDT/USD, the models were able to perform much better with accuracy of around 60%. Figure 4.18 and 4.19 show the confusion matrix and classification report for BDT/USD pair. We can see that the model is relatively biased towards the negative class and this could be attributed to the fact that the BDT/USD forex’s value fell more often. However, the classification report suggested that the model still maintained an acceptable recall rate even for the positive class.

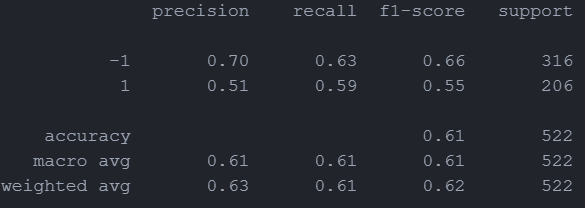
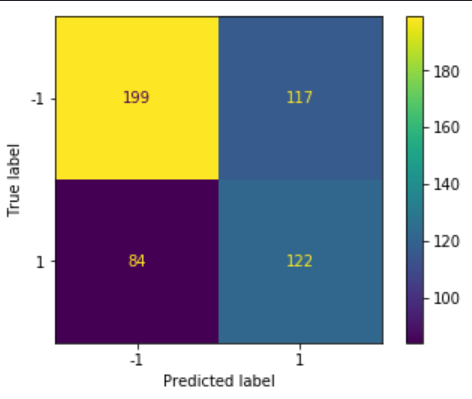


Figure 4.18 Confusion Matrix For Ridge Classifier for BDT

Figure 4.19 Classification Report for Ridge Classifier for BDT

Through the process of experimentation, SVC and Ridge Classifier were identified as high performing models on our given dataset. Classification into bins appeared to be a better approach to our method while making a smaller tradeoff for the scale to which we predict. Therefore, SVC and ridge classifier were chosen for optimization and further experimentation. Moreover, binary classification was chosen as the classification target variable.

## 4.4 Optimization

After conclusion of experimentation phase, the chosen high performing market indices and forex pairs were chosen as input for the optimization phase. The Experimentation identified SVC and Ridge Classifier as candidates for further optimization using the techniques discussed in Methodology.

### 4.4.1 Hyperparameter selection

Multithreading was used to take advantage of parallel computing and test a range of hyperparameters for our chosen model and their effects on model’s performance. A range of hyper parameters were tested. For SVC, hyperparameters including the regularization parameter, gamma, tolerance were tested along with different kinds of kernels that would fit the data better. The aim was to find a set of parameters that would build a model with low bias and high variance I order to reduce misclassification rate. In the case of Ridge Classifier, alpha (regularization parameter), tolerance and random state were experimented with.

Table 3 Best Set of Hyperparameters Identified on an 80-20 split for BDT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Regularization Parameter | Tolerance | Cross Market | Accuracy |
| Ridge Classifier | 0.21 | 0.006 | VND | 61.87 |
| Support Vector Classifier | 0.001 | 0.006 | IDX Composite | 61.1 |

Figure 4.20 shows how the accuracy for Ridge Classifier changes with respect to tolerance and alpha. It is interesting to saw how changes in alpha have a more evident effect on the performance of ridge classifier. From this we can infer that the chosen values for tolerance for the purpose of this graph were too small to create any noticeable effect when coupled with a low high. As alpha decreases or regularization increases, the accuracy of Ridge classifier increases. High regularization helps prevent overfitting of the model and could therefore be the reason behind improved performance. Since the peak is observed with low alpha and low tolerance, we ca conclude that a low alpha and tolerance would lead to more variance in the trained model.

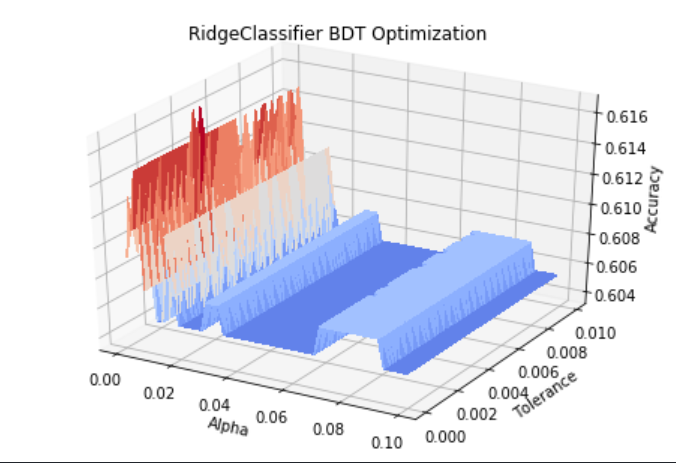


Figure 4.20 BDT Ridge Classifier Optimization

Through this process of optimization taking advantage of multithreading, we were able to identify set of hyperparameters that would be optimal for each chosen model.

### 4.4.2 Sliding Window Approach

It was hypothesised in the previous steps that the relatively poor performance of classification can be attributed to the inability of classification models to capture the trends in market effectively. Therefore, to help the model learn the trends more closely and make better predictions, a walk forward or sliding window approach was taken. The nature of time series data make it impossible for classification models to learn enough information from the data to accurate predictions far in the future. Moreover, it has been shown that the correlation between historic market data and current data decreases at an exponential rate. Moreover, results from ADF test had shown that smaller lag was significant in prediction. Finally, research had also shown that such a method is superior to a regular single train test split [21].

The walk forward approach uses a rolling window to make predictions. For each iteration a small number of datapoints are used to train the model. The model then predicts the next day’s value which is recording as our prediction. The rolling window size can be adjusted to investigate performance with training data of different sizes. It was suggested in (**Same citations)** that the next day’s market movement are highly correlated with how the market moved in the near past. Therefore, rolling widow sizes were kept relatively small (50 datapoints for data of 2600 days). Figure 4.21 compares the accuracy across different window sizes. It also shows the best accuracy achieved in an 80-20 split for reference. For training of these models, best hyperparameters were supplied as input.

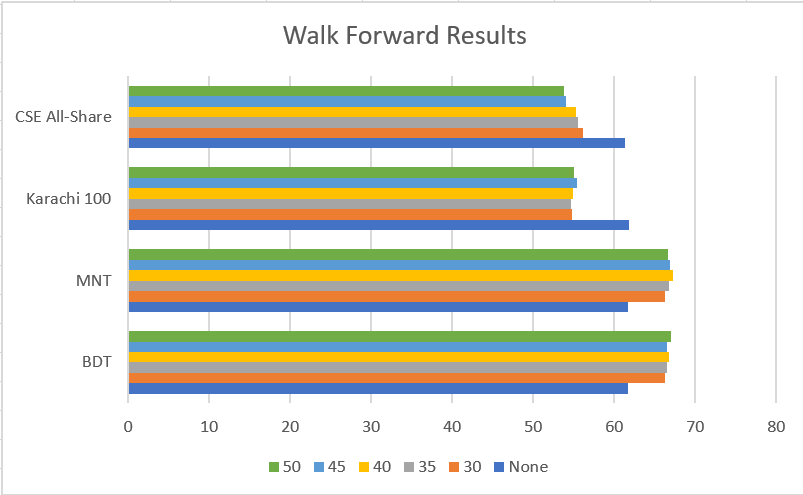


Figure 4.21 Accuracy Results Across Different window Sizes

Analyses of the results suggest that walk forward approach performs significantly better for forex. However, the optimal window period was not found for market indices. A different correlation exists between market indices and window period which needs to be investigated further. Despite this shortfall, walk forward approach performed better in average and was taken the optimal method for predicting forex rates.

Walk forward approach was also subject to optimization. The models were tested against several sets of hyperparameters to find the set which would lead to a better accuracy.

### 4.4.3 Ensemble Voting

The ensemble voting method is a simple addition to the current machine learning models. To take advantage of several models developed throughout the project and their predictive capabilities, an ensemble voting approach to prediction was adopted. In this approach, results from best performing models from the previous steps were used to obtain one result for each market index or forex.

In this approach, a simple mode was taken over all the predicted data produced by the different models identified as optimum for forex and indices. For forex, a mode was taken over all the results produced by the walk forward approach, resulting in a final prediction which takes advantage of different predictive capabilities of different window sizes. This increases the variance of the overall result and decreases the bias. Finally, for indices, results of classic classification models with optimized hyperparameters were chosen. Figure 4.22 shows results from ensemble voting approach. As it is evident, taking the most predicted class label from all the different predictions available lead to an overall improved performance, albeit marginal. The method was thus seen as he final optimal approach to predict direction of market movement for each of the chosen market index and currency.

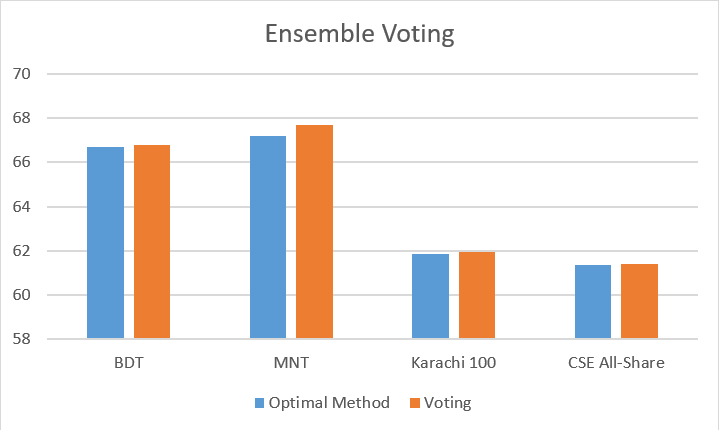


Figure 4.22 Ensemble Voting Results

## 4.5 Backtesting Strategies

Finally, the quality of the predicted data was analysed by testing the performance of our predictions against real market values to verify if they would still be profitable. Backtesting is a common technique to qualitatively analyse the predictions. For the purpose of this project, two backtesting strategies were developed with the help of pre-existing modules that provided several functionalities. Model performance was analysed by loading predictions on a strategy and analysing the final equity after a two-year period. The following subsections elaborate on the two approaches adopted.

### 4.5.1 Simple Moving Average Strategy

Following the implementations discussed in section 3.5.5, a simple moving average strategy was implemented. Since binary classification provide discrete values rather than continuous values, the binary prediction results were converted into a range of values by taking a rolling sum over n days resembling the approach in ensemble voting. Furthermore, slow and fast-moving averages were computed. A crossover of these moving averages would signal a trade. However, in order to be more precise and judge the prediction quality, prediction for time t+1 was also incorporated in the signal generation process. Now only if both the conditions were favourable, a trade was executed. If the slow-moving average would cross over a fast moving average the model predicts that the day would witness market closing at a lower price, held shares would be sold to save equity. In the other case, if the fast -moving average would surpass the slow-moving average, indicating an upward trend, a buy signal would be implemented if the model suggests that the day would close higher as well. Following Figures show a plot on equity over the 2-year backtesting period with a 0.2% commission. Optimizes window sizes used for final results are given in table 4.

Table 4 Optimized Window Period for Initial Strategy

|  |  |  |  |
| --- | --- | --- | --- |
| Market | Rolling Window for Sum | Slow Moving Average Window | Fast Moving Average Window |
| BDT | 6 | 5 | 20 |
| MNT | 2 | 10 | 20 |

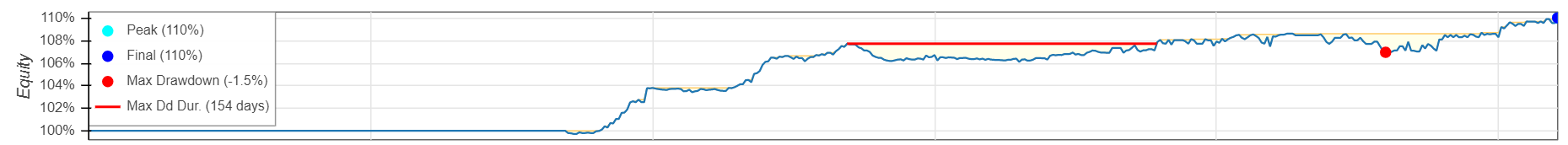


Figure 4.23 Final Equity Graph for MNT

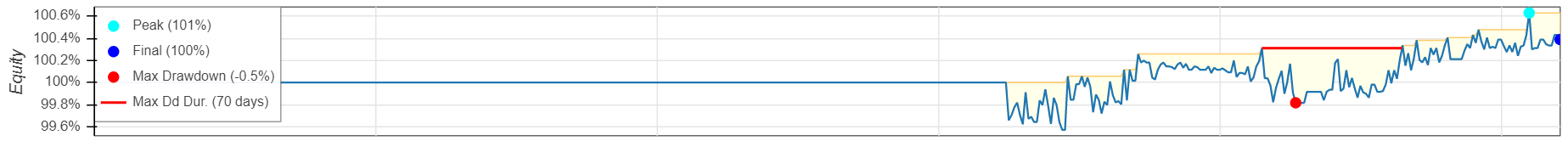


Figure 4.24 Final Equity Graph for BDT

This strategy would trade relatively infrequently and therefore another set of experiments were conducted without the additional constraint of the next day’s prediction following the trend predicted by the moving averages. The final equity here increased from 100% to 104% for BDT and increased from 100% to 114% for MNT. Optimized results of the strategy are provided in the table below. The following figures plot the changes in equity over the testing period with optimal window periods to maximize equity gain.

Table 5 Optimized Window Sizes for the New Strategy

|  |  |  |  |
| --- | --- | --- | --- |
| Market | Rolling Window for Sum | Slow Moving Average Window | Fast Moving Average Window |
| BDT | 2 | 10 | 20 |
| MNT | 9 | 25 | 75 |

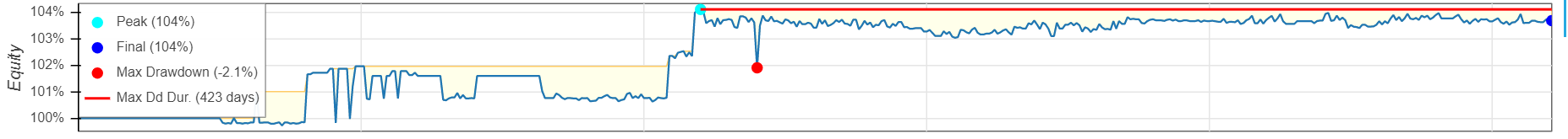


Figure 4.25 Changes in Equity for BDT With Less Constraints

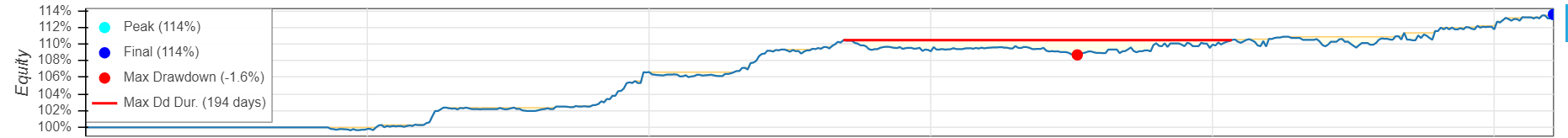


Figure 4.26 Changes in Equity for MNT With Less Constraints

### 4.5.2 Intraday Trading Strategy

To further analyses prediction quality and evade the dependence on our own historic predictions, an intraday market trading strategy was developed. The aim of this approach is to look at the prediction for the day for which the trading is to be conducted. Therefore, this strategy solely relies on the prediction of our model and takes advantage of walk forward prediction’s ability to predict data in line with recent trends.

The strategy was implemented over the last two years of the data collected. 500 shares were sold or bought with every trade. At every trade signal, the strategy would either short sell or long buy depending on the sign of the signal as explained in section 3.5.5. The initial euity was set as $100,000 and an average trading fees of 0.2% was set as commission. The following figures show changes in equity over the testing period for BDT and MNT.

As shown in figure 4.27 and 4.28, the strategy yielded a positive return in equity. The growth in equity is proportional to the changes in close price. This is the reason behind the significant different in final equity for both markets. However, it can be concluded that the even our model walk forward implementation to create better predictions fell prey to high volatility and shocks in BDT. This can be seen with a drop in equity after the shock periods as our model still predicted a growth in close prices even though the market was predominantly experiencing falls, But, the model quickly catches on to the trend and equity begins to grow again as the volatility of the market decreases. However, the final equity here is only 103% as opposed to best seen final equity of 104% in the previous strategy for BDT. However, for MNT, a 60% increase in equity was observed which is significantly better than the previous strategy.

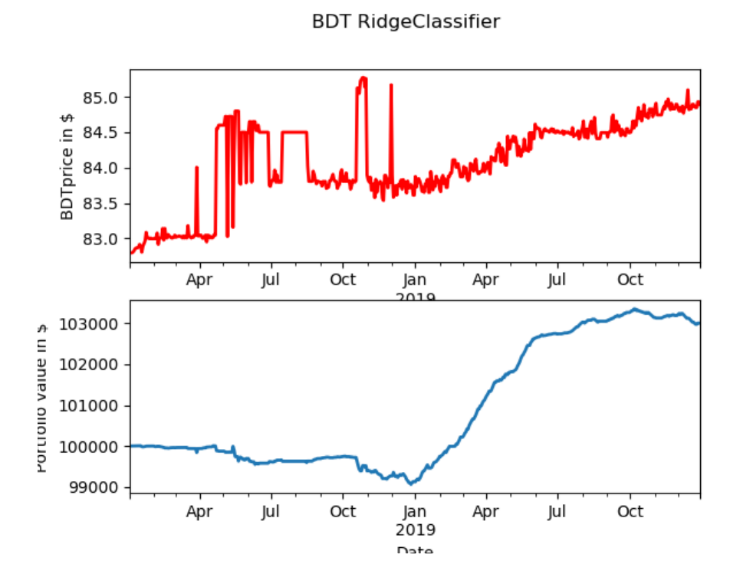
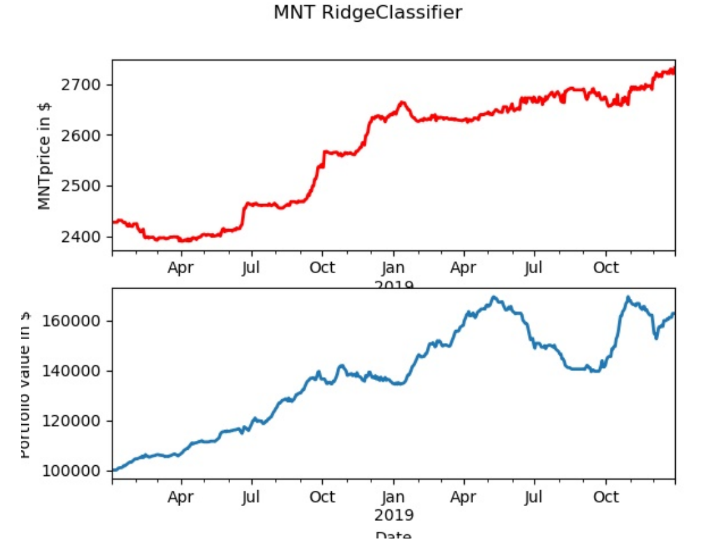
Overall results produced by the intraday market trading strategy were consistently better than those of moving average strategy. Therefore, along with capturing the general trend of the market, the predictions were also able to produce significant results on their own.

Figure 4.27 Changes in for BDT

Figure 4.28 Changes in Equity for MNT

## 4.6 Summary

This section discussed the various experiments conducted during this project and their major results. Firstly, numerical and textual data was collected and processed to act as input for the machine learning models developed and analyzed later. This step focused on setting up the codebase to scrape the internet for textual data to be later converted in market sentiment and collect numeric data that would represent the raw values of the traded commodity over a 10 years period. Textual data was then processed using BERT to represent market sentiment, a major indicator of market performance. The numeric data was subject data cleaning and interpolation to fill in missing data. Machine learning experimentation took place in an iterative process in conjunction with data pre-processing and model optimization.

The results from initial experimentation with regression models highlighted the need for to stationarity tests, the results of which suggested the need of derived features to act as input. Subsequently performance of model with these inputs were compared with performance using raw values which suggested that a stationarity significantly improves performances of machine learning models in time series data. However, despite improvements, the results from regression models were still unsatisfactory. The results from time series regression models tested were also unsatisfactory. This led to the shift in focus towards classification models.

K-bins classifier was developed to classify returns into k bins. However, even after optimization, the best performing model did not perform satisfactorily. The focus of the project was then narrowed down into binary bins classification. Initial analyses of binary classification suggested an improved performance. Despite high volatility in testing data, binary classification models could predict with 60% accuracy without optimization for forex markets and around 56% for indices. To best focus team’s resources, the best performing models, Support Vector Machine Classifier and Ridge Classifier were chosen for further optimization and evaluation. Initial hyperparameter optimization showed a significant performance increase. Both Ridge and SVM classifiers could now classify data with 61.6% accuracy.

To further improve on model performance, a walk forward approach was adopted. A rolling window of testing data would be used to predict the next day’s classification. By simply doing so, the model accuracy jumped to 67% for BDT. This is a significant increase an suggested that a short window period is optimum for forex prediction. However, the same approach on indices showed decreased performance suggesting even a window period of 50 days was not enough to have improvements in prediction. Furthermore, ensemble voting was carried out over the results of several models developed to get the most probably classification. This showed an increase in performance, albeit minor.

Models were only analyzed quantitatively so far and therefore, to qualitatively analyze the models’ predictions, backtesting strategies were developed. The results from the optimized models were subject to strategies based on simple moving average and intraday trading. In both cases, the predictions were able to produce gains in equity over a two year period. The first strategy analyzed the trend capturing capabilities of the models and the second judged the quality of predictions. The positive final equity showed that the models were able to bot identify trends and produce decent results.

# **CONCLUSIONS**

Successful prediction of financial instruments can yield significant profits. After thorough analyses of literature available on this topic, we identified that there is lack of research that particularly deals with employing cross domain analysis using neural networks for predictions models. Therefore, this project aimed to compare performance of several machine learning models. It achieved this by providing a set of features derived from raw financial values and carried out an inclusion-exclusion approach to test for appropriate additional features like sentiment scores and cross market features.

## 5.1 Major Conclusions and Results

We identified the importance of discovering the hidden relationships between various parameters and input data. Correctly detecting these relationships are imperative to the success of this project. Moreover, effective feature extraction can aid in the process of developing machine learning models with higher accuracy. Therefore, an iterative process of data analysis, feature extraction and building models was adopted in this project. This was supplemented with additional optimization techniques to gather more insights and take advantage of the predictions generated.

### 5.1.1 Conclusions from Machine Learning Models

The experimentation phase concluded with identification of suitable model and better approaches to be carried forward into the optimization steps. The conclusions and insights gained from this phase are provided below.

Both regression and time series models failed to provide significant results prompting the team to explore classification models in more depth. This led to increased performance and improved accuracy. Binary classification method was chosen as the best performing in classification due to consistently higher accuracy rates as compared to K-bins classification. Out of the many models testing, the team narrowed down the scope to Support Vector Machine and Ridge Classifier as our final models. As per our investigation, Support Vector Machine Classifier’s performance can be accounted to several kernels that can intercept complex data and create high dimension decision boundaries. Moreover, SVM avoids the risk of overfitting by incorporating a slack variable that allows flexibility of the decision boundary. Similarly, Ridge Classifier incorporates an aspect of regularisation which is theoretically similar to the slack variable in SVM. In both cases, the additional parameter constraints the weights and prevents overfitting. These models were thus selected and further optimized to better fit the data in the optimization stage.

For all models and methods, cross validation was applied on training sample. This step involved incrementally adding time series split to the validation test during model training. However, this did not result in any significant noticeable and was therefore not considered to not have any advantage. However, this approach acted as base to develop a walk forward implementation.

The experimentation phase highlighted the importance of stationarity in time series data and how it can influence the prediction capabilities of a model. It was seen that predicting raw values leading to insignificant results suggesting that a model could not learn any important information from the data and would generally overfit leading to high bias and low variance. Therefore, the data was made stationary by calculating derived features and moving average and lagged values of feature variables was provided as input. The target variable was also shifted to close return, leading to significant jump in performance both in regression and classification models.

Initial data analysis and exploration had suggested a correlation between several pairs of markets. Highly correlated forex and market index was therefore also included as an input feature (See table 1). Such inclusion of cross domain features led to a high feature space and therefore feature extraction was carried out during cross domain analysis. This was done with the help of PCA algorithm. The algorithm’s results suggested that 10 features explained 98% of the variance and therefore the top 10 variables were chosen to reduce dimensionality and improve model fit time. PCA only led to a marginal difference in accuracy but was preferred overall as it would significantly reduce training and testing times during optimisation.

Finally, sentiment scores were also incorporated as features while testing. Experiments were conducted using the inclusion exclusion principle. The results obtained showed no improvements in results. Further investigation suggested that sentiment score was not highly correlated with the target variable (close returns) for any of the chosen markets. Furthermore, sentiment analyses did not provide any additional advantage in predicting shocks in the market. The sentiment score varied uniformly around the mean and did not show any obvious trend.

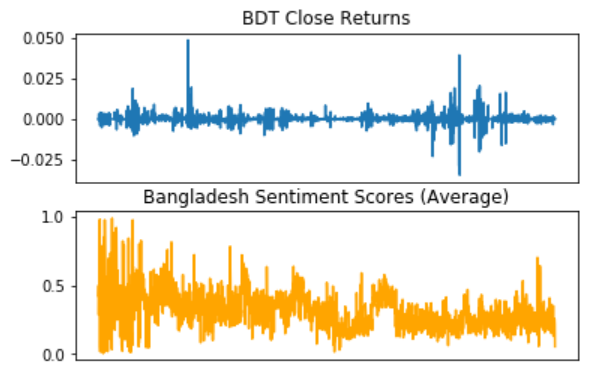


Figure 5.1 Sentiment Score for Dhaka Compared with BDT Close Return

It is hypothesised that market sentiment was already reflected in the market data and adding sentiment score did not provide any additional insights. Any political or economic decision that would have a significant effect in market performance was generally already reflected in the market before public had a chance to react to it. This is explained as part of the Efficient Market hypothesis which states that the price on an asset already reflects all available information related to it.

**5.1.2 Conclusions from Optimizations and Backtesting**

Through optimization, several fallacies in our initial experimentation were identified. This included the lack of specific hyperparameters during initial testing. Therefore, inclusion of hyperparameters selection showed a considerable jump in accuracy. It was seen that a small regularization parameter significantly increases accuracy by increasing penalty on misclassification. Furthermore, a decreased tolerance would allow for more support vectors to be identified and a better margin could be constructed. All three forms of optimization carried out produced improved results with cross market features. Optimization techniques also solidified our claims that cross market, whether intra market or inter market, generally produce better results than without any additional features. Ensemble voting with an 80-20 split was identified as the best method for prediction for indices if accuracy was of importance. For forex, ensemble voting with results from sliding window approach as input was chosen as the best method for prediction.

Results from all three optimization techniques were backtested. Highest performance was observed with results from ensemble voting optimization approach. As shown in the figures, the results from this model were both able to identify the trends in market and produced prediction that would lead to gain in equity even on simple trading strategies.

Currently, we have concluded historic data collection and are just collecting data on a daily basis to update our database with most recent information. In future, we plan to conduct exploratory data analysis on the data collected so far. We aim to identify some trends that can aid us in the upcoming model development process. We will continue collecting data daily to grow our database for better training and testing of or models. We will then begin testing open source models by fine-tuning parameters to make satisfactory predictions. In the upcoming few weeks, we will set up a dashboard for the results achieved up till then and publish them on our website. This process will repeat itself with newer models in each iteration. In later stages of this project, we aim to work towards building a new model that can be used to make financial predictions.

## 5.2 Further Research

After obtaining considerable results with almost 70% accuracy with the best methods, the team hypothesises that these approaches can further be improved to improve usability and prediction accuracy. These suggestion to future research is highlighted in the subsections below.

Support vector machines are one of the most powerful ensemble learning machines. They provide an option to provide custom kernals which could be used to better model the data. The team hopes to delve further into analysing the data and hoping to develop a kernal for SVM that would produce more optimal results by factoring in exponential smoothing developed by holts winters.

Furthermore, the models currently developed can be analysed on out of box data by adding support for online learning and prediction. This could give more insights into the model’s variance when unforeseeable economic trends take place. By scaling the project and incorporating online learning, the team believes that there is potential to generate relatively accurate predictions which could be used in trading.

## 5.4 Work Distribution

The workload was distributed between the two team members of this project. The table below shows the work distribution for the project

Table 6 Division of Work

|  |  |
| --- | --- |
| Shubhankar Agrawal | Karan Mahajan |
| * Project Website * Data Collection (Financial data and social media data) * Financial data pre-processing, feature engineering and Exploratory Data Analysis * Model Experimentation phases | * Performing Sentiment Analysis on collected social media data * Textual data pre-processing – Aggregating Sentiment scores * Model optimization techniques * Backtesting and Trading strategies |

# **REFERENCES**

[1] G. Zhang, Neural Networks in business forecasting. USA: IGI Publishing, 2004.

[2] S. P. X. Jiang, J. Jiang and G. Long, "Cross-Domain Deep Learning Approach For Multiple Financial Market Prediction," presented at the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, 2018.

[3] S. I. Ao, "A hybrid neural network cybernetic system for quantifying cross-market dynamics and business forecasting," Soft Computing, journal article vol. 15, no. 6, pp. 1041-1053, June 01 2011, doi: 10.1007/s00500-010- 0580-4.

[4] A. Ludhiyani, S. Joshi, R. Pathak, P. Parandkar, and S. Katiyal, "Quant techniques and regime analysis for increase in trade & business between India and Israel," in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015: IEEE, pp. 442-447.

[5] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[6] E. K. Laitinen and T. Laitinen, "Bankruptcy prediction: Application of the Taylor's expansion in logistic regression," International review of financial analysis, vol. 9, no. 4, pp. 327-349, 2000.

[7] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: a new approach," in 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005: IEEE, pp. 192-196.

[8] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis," Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf), vol. 15, 2012.

[9] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," Journal of computational science, vol. 2, no. 1, pp. 1-8, 2011.

[10] K. Zhang, L. Li, P. Li, and W. Teng, "Stock trend forecasting method based on sentiment analysis and system similarity model," in Proceedings of 2011 6th International Forum on Strategic Technology, 2011, vol. 2: IEEE, pp. 890-894.

[11] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, "Exploiting topic based twitter sentiment for stock prediction," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2013, pp. 24-29.

[12] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," PloS one, vol. 12, no. 7, p. e0180944, 2017.

[13] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in Twenty-fourth international joint conference on artificial intelligence, 2015.

[14] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, "Predicting stock market trends by recurrent deep neural networks," in Pacific rim international conference on artificial intelligence, 2014: Springer, pp. 759-769.

[15] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," Neurocomputing, vol. 137, pp. 47-56, 2014.

[16] R. Liu, "Correlation between fundamental values in the financial market [electronic resource]," Thesis (M.S.)-- Southern Illinois University at Carbondale, 2006., 2006.

[17] D. Yang, "The socioeconomic determinants of correlations between stock market returns as revealed by a gravity model," Concordia University, 2009.

[18] G. Box, G. Jenkins, and G. Reinsel, "Time Series Analysis- Forecasting and Control, ; Prentice-Hall: Englewood Cliffs, NJ, 1994."

[19] U. Thissen, R. van Brakel, A. P. de Weijer, W. J. Melssen, and L. M. C. Buydens, "Using support vector machines for time series prediction," Chemometrics and Intelligent Laboratory Systems, vol. 69, no. 1-2, pp. 35-49, 2003, doi: 10.1016/S0169-7439(03)00111-4.

[20] A. Widodo, I. Budi, and B. Widjaja, "Automatic lag selection in time series forecasting using multiple kernel learning," Int. J. Mach. Learn. & Cyber., vol. 7, no. 1, pp. 95-110, 2016, doi: 10.1007/s13042-015-0409-7.

[21] K. Żbikowski, "Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy," Expert Systems with Applications, vol. 42, no. 4, pp. 1797-1805, 2015.